

Selector de fruta y simulación de una aplicación real



Alumna: Miriam Fernández Ribot

Tutor: Javier Ruiz Hidalgo

Categoría: Proyecto Final de Carrera

Fecha de entrega: 08/06/2006

Índice

1. Introducción.....	3
2. Estado del Arte.....	5
2.1. “Sistema de Clasificación automática de fruta utilizando procesado de imagen”	5
2.2. “Sistema inteligente de selección de fruta basado en el estudio del color con procesado de imagen”	6
2.3. “Clasificación por calidad de naranjas basado en visión de alta velocidad”	7
2.4. ” Desarrollo de una máquina con sistema de visión en tiempo real para un robot recolector de manzanas”	8
3. Consideraciones previas.	9
3.1. Software.....	9
3.2. Adquisición y adaptación de imágenes.....	10
4. Descripción del sistema.....	13
4.1. Segmentación.....	15
4.1.1. Selección del espacio de color.....	16
4.1.2. <i>Thresholding</i> . Determinación de umbrales.....	18
4.1.3. Uso de operadores morfológicos. Apertura o cierre.....	22
4.2. Detección de contornos.....	31
4.3. Cómputo de contenedores en histogramas.....	43
4.4. Análisis de la mediana en HSV.....	49
5. Resultados y observaciones.....	59
6. Conclusiones.....	73
7. Anexos	76
7.1. Situación del algoritmo “selector” en la cadena logística.....	76
7.2. Sistemas de adquisición en la cadena logística.....	79
8. Referencias bibliográficas.....	81

1. Introducción

La idea del proyecto es la de generar un sistema de detección de fruta en mal estado. Para ello se utilizarán técnicas simples de procesamiento de imagen tales como histogramas, uso de operadores morfológicos, cambios a otros espacios de color entre otras.

Cualquier persona que tenga contacto con el mundo de la agricultura puede darse cuenta de cuán duro es su trabajo. Plantan, recolectan la fruta y la clasifican para posteriormente enviarla a empresas logísticas que empaquetan la fruta y la trasladan a los centros comerciales y fruterías para su consumo. Dentro de estas empresas un cierto número de trabajadores se dedica, exclusivamente, a clasificar la fruta que les llega y decidir si es apta o no para el consumo directo o bien si debe ir destinada a zumos: el color, los defectos, el tamaño, etc. Todo se tiene en cuenta.

Este proceso manual, hoy en día está siendo sustituido por maquinaria especializada que realiza la tarea de estos trabajadores mediante técnicas de procesamiento de imagen. Sin embargo hasta la fecha solamente unas pocas empresas disponen de este tipo de aparatos, que se incluyen a la cadena de limpieza y embalaje, ya que resultan caros.

El estudio que se detalla a continuación se basa en un programa realizado con Matlab® de The MathWorks que simulará, a partir de imágenes de entrada, el procedimiento que podría realizar un aparato seleccionador para determinar la calidad de la pieza de fruta. No se analizará en ningún momento la parte mecánica del tema sino que sencillamente se describirán una serie de pasos basados en procedimientos de procesamiento digital de imagen que determinarán si la fruta a analizar está en buen estado o no, procurando determinar qué tipo de defecto se detecta.

Las aplicaciones de este tipo de sistema selector de fruta pueden ser diversas: desde una máquina incorporada a una cadena de seleccionadora y envasadora de fruta en almacenes logísticos a un aparato para un departamento que quiera

gestionar el tipo de defectos en la fruta a partir de una muestra y poder llevar un control estadístico de zonas geográficas (control de plagas, insectos causantes de la plaga, pesticidas a utilizar, etc. pasando también por aquellas empresas aseguradoras de agricultores que podrían realizar una base de datos a partir de la clasificación de defectos (sería posible relacionar la causa de una cosecha en mal estado con el análisis de defectos en la fruta, como el viento, granizo, etc. con los pagos a los agricultores propietarios del seguro). También sería interesante para estudiar los tipos de plagas existentes y sus efectos en la fruta o bien para estudiar casos nuevos de insectos no autóctonos y qué clase de daño crea en la fruta de la zona.

Sin embargo, a pesar de todas estas posibles aplicaciones, hoy en día la más interesante y solicitada sigue siendo la de llevar a cabo la selección que se suele hacer manualmente y agilizar la cadena logística de la empresa para poder, de este modo, distribuir más rápido y mejorar los tiempos de distribución.

2. Estado del Arte

En la Universidad Politécnica de Catalunya no se han encontrado referencias de proyectos similares, pero es cierto que se han realizado y se llevan a cabo hoy en día estudios parecidos. Es el caso del Centro de Visión por computadora de la Universidad Autónoma de Barcelona (UAB), donde aplican técnicas de visión por ordenador para la clasificación de manzanas. Este centro estuvo presente en el espacio FITEC de la feria *Alimentària 2006* que tuvo lugar en Barcelona a principios del mes de marzo y donde expusieron las ventajas de este tipo de técnica. [3]

También son conocidos los estudios llevados a término por investigadores japoneses y chinos así como también americanos y de Nueva Zelanda.

En los siguientes puntos se hace referencia a los proyectos parecidos encontrados en Japón, la China, el Reino Unido e Italia.

(Títulos traducidos al castellano)

2.1 “Sistema de Clasificación automática de fruta utilizando procesado de imagen”

Este estudio, llevado a término por John B. Njoroge. Kazunori Ninomiya. Naoshi Kondo y Hideki Toita del Departamento de Desarrollo Tecnológico de Japón y publicado el año 2002 utiliza técnicas de procesado de imagen digital para mejorar el transporte y clasificación de la fruta.

Su sistema se basa en la inspección interna y externa de la fruta ya que se analiza tanto el color del fruto como su contenido en ácidos y azúcar. Las imágenes obtenidas a partir de tres cámaras son procesadas por tres ordenadores diferentes. Asimismo disponen de un aparato de Rayos-X para detectar los defectos biológicos de la fruta y varios sensores que analizan azúcares y ácidos.

La detección de defectos se hace analizando el nivel de componente azul de las imágenes tomadas, comparándolo con las componentes sumadas R, G y B y siempre teniendo en cuenta la iluminación en las imágenes.

Para este estudio los investigadores han utilizado naranjas, ya que suponen el 40% de la producción de fruta al Japón y cualquier avance en la agilización del proceso de almacenaje, clasificación y distribución significa una mejora en la eficiencia del proceso.

2.2 “Sistema inteligente de selección de fruta basado en el estudio del color con procesamiento de imagen”

El estudio que lleva por nombre el título de este apartado ha sido realizado por Guo Feng y Cao Qixin, del Instituto de Investigación de Robótica de la universidad Jiao Tong de Shanghai, China.

Sus estudios se han orientado hacia la segmentación de la fruta, la eliminación de ruidos y la detección de contornos.

El primer paso de la técnica aplicada es segmentar la imagen aplicando una binarización (estableciendo un umbral que determina si el píxel a tratar será blanco o negro) sobre la imagen convertida a espacio de color Ohta (de RGB a Ohta). El cambio de RGB a Ohta es lineal, cosa que no sucede con las conversiones a HSV (*), HSL (**), etc.

El espacio de color Ohta se obtuvo analizando más de 100 características distintivas que se obtuvieron durante la segmentación de ocho tipos de imágenes en color. Se logró encontrar un conjunto de características ortogonales de color que se llamaron a partir de entonces “espacio de color Ohta”. Este espacio de color casi no tiene coste computacional y su resultado es bastante satisfactorio. Las expresiones de transformación de RGB a Ohta son las siguientes:

$$\begin{cases} I1 = (R+B+G)/3 \\ I2 = (R-B)/2 \\ I3 = (2G-R-B)/4 \end{cases} ; \begin{cases} I2' = R-B \\ I3' = (2G-R-B)/2 \end{cases}$$

Después se aplica un algoritmo especial que elimina ruido en la imagen binaria para después aplicar un algoritmo de detección de contornos y así poder segmentar la imagen de manera robusta. Este algoritmo denominado “Spline-Interpolation Algorithm” calcula las coordenadas geométricas del centro de la manzana de la imagen sin ruido y se busca veinticuatro puntos del contorno del fruto con una diferencia de 15° cada uno. Seguidamente se calcula la distancia entre los puntos de contorno y el centro geométrico y se utilizan las medidas como longitudes de unos radios imaginarios. Con estos datos el algoritmo de Spline-Interpolation genera un contorno suave de la manzana.

Finalmente se realiza una transformación a HSV, utilizando la imagen H para determinar un margen de color admisible necesario para establecer criterios sobre la calidad de la fruta.

2.3 “Clasificación por calidad de naranjas basado en visión de alta velocidad”

El estudio que se explica a continuación ha sido realizado por Michael Recce y John Taylor del Departamento de Anatomía y Biología de Desarrollo del Colegio Universitario de Londres (Londres, Reino Unido) en conjunción con Alessio Plebe y Giuseppe Tropiano de l’ A.I.D en Italia.

En este estudio se utilizan las naranjas: cómo analizarlas y detectar el tallo en las imágenes aplicando procesamiento digital y una búsqueda de defectos locales. Este aspecto es interesante porque a menudo el tallo de la fruta se confunde con un defecto.

Inicialmente la técnica que utilizan es la generación de histogramas de componente rojo (R) y verde (G) para clasificar el color (el color determina el

tipo de naranja, en este caso, debido a la naturaleza del fruto) mirando la distribución de los píxeles de la fruta. Los defectos y el tallo acostumbran a ser distorsiones en el histograma y se pueden comparar con una distribución normal según un histograma de una naranja “sana”.

Finalmente el grueso de la investigación se centra en la búsqueda de defectos en la superficie de la fruta, segmentando las regiones defectuosas de aquellas que no lo son utilizando máscaras y los mismos histogramas.

La marca del tallo en cualquier fruto suele confundirse con un defecto en el momento de procesar la imagen. Los creadores de este estudio han basado la diferenciación del tallo respecto un defecto cualquiera con el hecho que el tallo es una zona con más textura y con simetría radial.

2.4 ” Desarrollo de una máquina con sistema de visión en tiempo real para un robot recolector de manzanas”

El estudio realizado por D.M Bulanon, T. Kataoka, H. Ukamoto y S. Hata de la Escuela de Graduación en Agricultura, de la Universidad de Hokkaido en Sapporo, Japón, es muy completo puesto que el objetivo final es la creación de un robot que recoja las manzanas directamente del árbol. La parte de procesado de imagen que incluye este estudio incorpora explicaciones sobre cómo se distingue el fruto del resto del árbol; es decir, cómo decide el aparato aquello que debe recolectar. Inicialmente hace falta decir que el robot incorpora un sistema de visión que primero adquiere la imagen, gradúa la luminosidad y realiza un análisis y procesado digital de la imagen.

El primer paso en el análisis de la imagen es la separación de la fruta y el árbol utilizando una conversión de RGB a coordenadas cromáticas:

$$\begin{aligned} r &= R / (R+G+B) \\ g &= G / (R+G+B) \\ b &= B / (R+G+B) \end{aligned}$$

Figura3

Para la realización de la segmentación se utiliza una técnica denominada “decisión - acercamiento teórico” que consiste en una función que discrimina valores o parámetros según el reconocimiento de un cierto patrón: esta técnica actuará de una u otra manera en función de lo que detecta.

Tras segmentar la fruta se procede a aplicar uno filtro paso bajo para eliminar el ruido de la imagen.

El robot no detecta defectos, pero sí es capaz de discernir entre fruta y resto de árbol: follaje, ramas, etc.

3. Consideraciones previas.

Antes de proceder a la descripción del sistema es necesario tener en cuenta una serie de consideraciones, tales como el software a usar y qué imágenes serán necesarias para el selector. También es remarcable que los procedimientos que se aplican para el procesado digital de imagen son diversos y un gran número de veces el escoger uno u otro se hace de forma subjetiva. Sin embargo en el punto de descripción del sistema se expondrán las diversas opciones y las decisiones tomadas durante el estudio.

3.1 Software.

Para realizar el algoritmo que determina la selección de la fruta es necesario escoger antes un entorno de trabajo que cómodo y ofrezca las herramientas necesarias para programar. Es por este motivo que se ha escogido como software Matlab® ya que ofrece un entorno de trabajo sencillo de utilizar

Matlab® permite la programación de alto nivel y posee un amplio conjunto de funciones preprogramadas que facilitan el trabajo con imágenes: filtros, funciones de morfología y algoritmos diversos para la detección de contornos, reducción de ruidos y cambios de formatos de imagen (*Image Processing*

Toolbox de Matlab®). Este programa también incluye un extenso conjunto de ejemplos y demostraciones.

3.2 Adquisición y adaptación de imágenes.

Las frutas serán melocotones en este estudio y se han fotografiado desde diversos puntos de vista, obteniendo así un conjunto de imágenes que conformarán la base de datos necesaria para el “selector”. Es destacable el hecho que en un caso real una o varias cámaras tomarían las imágenes de un único melocotón, conformando un grupo de imágenes sobre el mismo elemento que serían analizadas posteriormente por separado.

Como es necesario obtener melocotones en mal estado para extraer aquellos parámetros que los diferencian de los sanos, se han tenido que recrear los defectos en la superficie dejando caer las piezas desde cierta altura, simulando roces que dejan entrever la pulpa y punzando la superficie de la piel.

Se ha escogido la variedad de melocotón “Flavor Crest” procedente de Fresno, California (EE.UU.) y creada el año 1974. Aún así, también se ha estudiado las variedades “May Crest”, “Candor” y “Rojo” para la elaboración de resultados, ya que las variedades en los establecimientos varían en función de la temporada.

Para hacer las fotografías se ha tenido en cuenta: fondo, resolución, iluminación y encuadre. La resolución establecida ha sido la máxima de la cámara Sony DSC 93-A (5 megapíxeles) y el encuadre manteniendo la fruta en medio de la imagen. Las condiciones de fondo e iluminación han sido complicadas. Una mala iluminación puede afectar todo el tratamiento de la imagen porque las sombras que aparecen pueden confundirse con regiones parecidas en grises del melocotón. La luz con que se han hecho las imágenes es la característica de un día soleado a la sombra usando el flash y sin activarlo. El fondo con que se ha tomado las imágenes es simple (blanco), La iluminación y el fondo son aspectos que se tratan más adelante.

Cuando finaliza la adquisición de imágenes es necesario escalarlas a un tamaño más cómodo para trabajar ya que con 5 megapíxeles el gasto computacional sería muy grande, siempre manteniendo la relación de aspecto inicial. De la resolución inicial se pasa a 640x480. Es importante remarcar que Matlab® trata las imágenes como matrices por lo que resulta conveniente el reescalado de las imágenes de entrada para agilizar también el tiempo de ejecución de las funciones.

El formato de las imágenes es JPG por defecto en la cámara fotográfica. Aunque se use JPG, los efectos típicos de esta compresión, como por ejemplo el efecto de bloque, no se aprecian al bajar tanto la resolución.

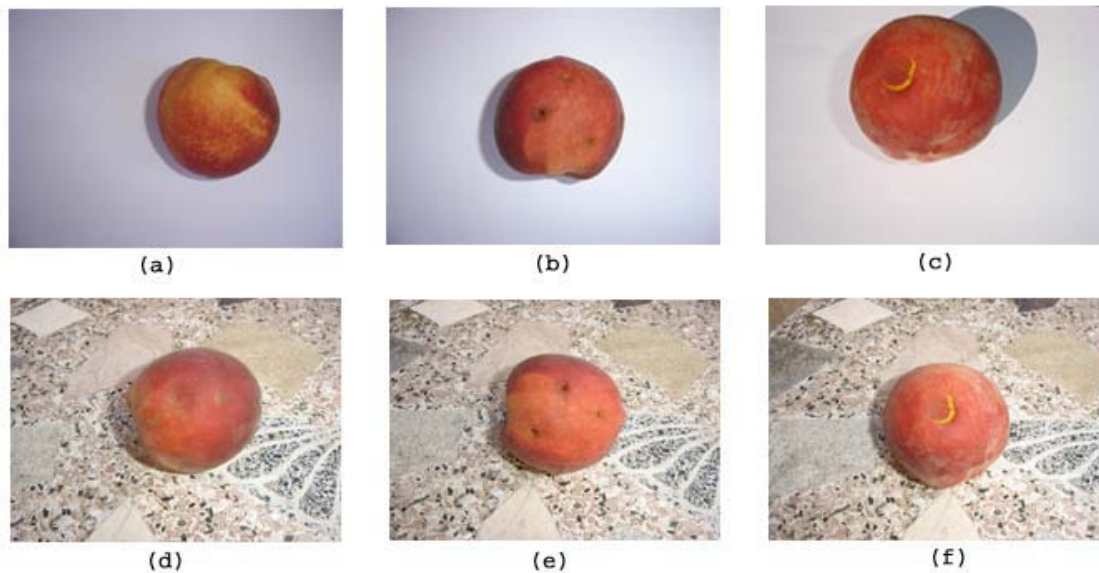


Figura 1 (a) Melocotón sano sobre fondo simple; (b) Melocotón picoteado sobre fondo simple; (c) Melocotón con pulpa visible sobre fondo simple; (d) Melocotón golpeado sobre fondo complejo; (e) Melocotón picoteado sobre fondo complejo (f) Melocotón con pulpa visible sobre fondo complejo.

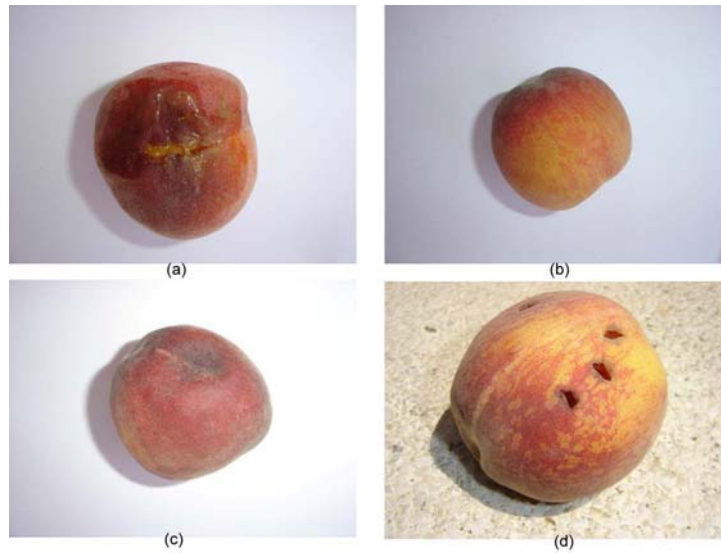


Figura 2 (a) Melocotón con pulpa visible sobre fondo simple (b) Melocotón picoteado sobre fondo simple; (c) Melocotón golpeado sobre fondo simple; (d) Melocotón picoteado sobre fondo con textura compleja.

4. Descripción del sistema

El esquema de la figura 4 muestra el recorrido de la imagen de la pieza de fruta hasta determinar si ésta es buena o mala.

Inicialmente se lee la imagen escalada a 640x480 en el Matlab® y se le asigna una variable. Ésta es una matriz de tamaño 640x480 píxeles en formato uint8. Por comodidad, la imagen es convertida de formato uint8 a double. La diferencia entre uno y otro es el rango de valores que tienen los píxeles. Para el caso uint8 los píxeles toman valores en el intervalo [0,255] mientras que para el caso double lo hacen entre [0,1].

El esquema se inicia en el punto en que la imagen del melocotón entra en el bloque de segmentación. La imagen resultante es el melocotón separado del fondo constando éste como valor uniforme negro. Posteriormente la imagen segmentada se introduce en un detector de contornos cuyo objetivo es obtener un valor que resulte decisivo y determine si el melocotón es apto para el consumo directo (si está sano o bueno) o bien tiene algún defecto.

En el proceso de almacenaje de melocotones en cajas en una cadena de un centro logístico frutero cualquiera solamente son escogidos los melocotones considerados sanos, mientras que aquellos que no son destinados a otros usos tales como la producción de zumos y mermeladas.

En el esquema propuesto se va más allá de la clasificación de “melocotón bueno” y “melocotón malo”: se trata de clasificar de forma simple el tipo de defecto en cuestión que hace descartar el fruto; a saber: golpes, roces que dejan ver la pulpa del melocotón y daños producidos por los picotazos de los pájaros .

Para determinar el tipo de defecto se usarán dos bloques: un contador de contenedores en histograma y otro donde se analizará la mediana de los histogramas de la imagen S de HSV.

Las diferentes partes que componen el algoritmo se detallan en los siguientes apartados.

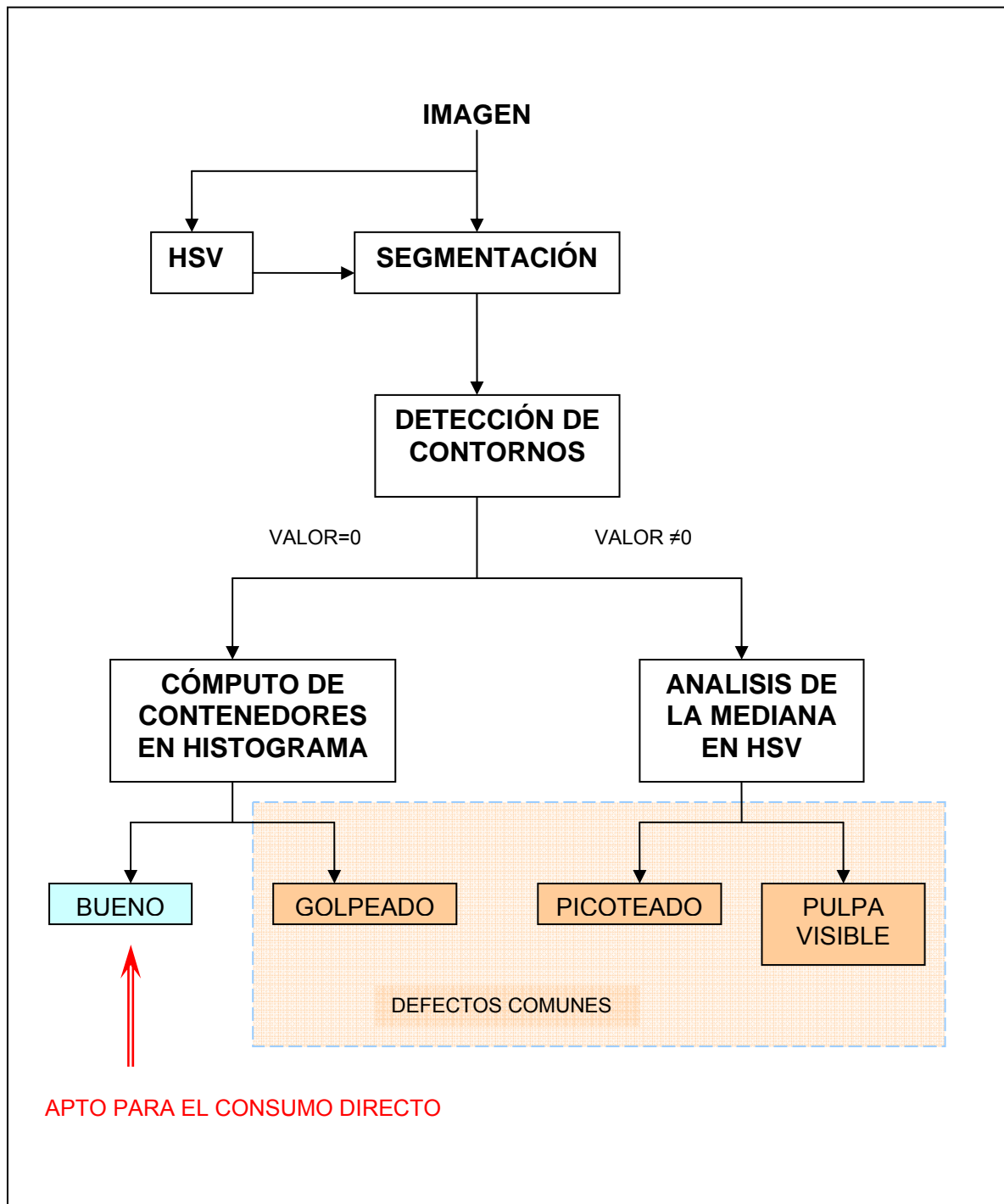


Figura 4 Esquema del sistema "Selector".

4.1 Segmentación

El primer paso en el análisis de imágenes generalmente es la segmentación de la imagen. Este procedimiento subdivide la imagen en las partes que la constituyen u objetos. El nivel en que se realiza la subdivisión depende del problema a resolver. Es decir, la segmentación debería pararse cuando los objetos de interés en una aplicación han sido aislados.[1][González, Woods,1993]

Después de adquirir y preparar las imágenes se procede a investigar las transformaciones más convenientes y obtener aquellos parámetros que pueden ser útiles para la elaboración del algoritmo. Antes, sin embargo, debe limitarse la zona de trabajo del programa al lugar de la imagen donde se localiza el melocotón, ya que éste y no el fondo es la zona de interés. Por este motivo se procede al estudio de técnicas de segmentación con el objetivo claro de “separar” la fruta del fondo de la imagen aceptando como válida una imagen segmentada que no excluya los defectos del melocotón. Se da por entendido que no es necesario que la segmentación sea perfecta, pues no es este el objetivo del proyecto, pero si debe ser suficiente para no excluir información.

Lo que se pretende conseguir es que el fondo muestre un valor homogéneo igual a 0 que equivale al negro, siendo 1 el valor de blanco que caracterice el objeto “melocotón”. Los procedimientos o técnicas establecidas para llevar a cabo la segmentación son diversos: la detección de discontinuidades en la imagen, el enlace de bordes y detección de límites, el *Thresholding* (o uso de umbrales) y finalmente la aplicación de técnicas de segmentación orientada a objetos. La utilización de una u otra técnica depende de la complejidad de la imagen a tratar, o mejor dicho, de la cantidad de objetos a segmentar.

Para el caso que plantea el proyecto y a este nivel del algoritmo no es necesaria una técnica complicada de segmentación porque sólo interesa la extracción del melocotón del fondo. El hecho de haber tomado las imágenes con un fondo sencillo simplifica el proceso de segmentación considerablemente

ya que no se corre el riesgo de obtener objetos segmentados aparte del melocotón y que serian texturas de fondo que no interesan.

4.1.1 Selección del espacio de color

Antes de proceder a la segmentación de la imagen se tiene que decidir el espacio de color desde el cual se trabajará y que debe ser congruente con los espacios que Matlab® permite: RGB, color indexado, HSV, niveles de gris, binaria...

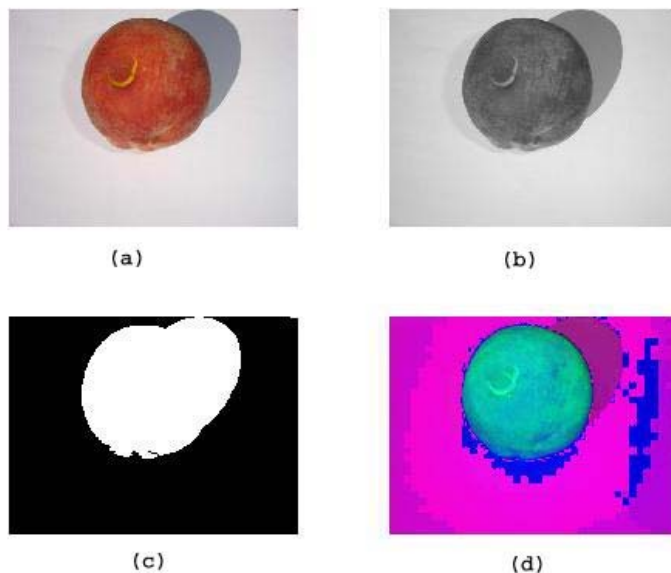


Figura 5 (a) Imagen original de la fruta (b) Fruta en niveles de gris
(c) Imagen binarizada con la sombra añadida al objeto de interés (d) Imagen HSV.

Después de realizar diversas transformaciones se comprueba que el modelo HSV es el más adecuado para el análisis de imágenes y su tratamiento. Aún así las imágenes en niveles de gris no se descartan para comprobaciones o análisis extras.

$$\begin{aligned}
 & \text{MAX}=\text{MAX}(\text{R},\text{G},\text{B}); \text{MIN}=\text{MIN}(\text{R},\text{G},\text{B}) \\
 & \text{H} = \begin{cases} 60 \times \frac{\text{G}-\text{B}}{\text{MAX}-\text{MIN}} & ; \text{ if MAX=R and G} \geq \text{B} \\ 60 \times \frac{\text{G}-\text{B}}{\text{MAX}-\text{MIN}} + 360 & ; \text{ if MAX=R and G} < \text{B} \\ 60 \times \frac{\text{B}-\text{R}}{\text{MAX}-\text{MIN}} + 120 & ; \text{ if MAX=G} \\ 60 \times \frac{\text{R}-\text{G}}{\text{MAX}-\text{MIN}} + 240 & ; \text{ if MAX=B} \end{cases} \\
 & \text{S} = \frac{\text{MAX}-\text{MIN}}{\text{MAX}} \\
 & \text{V} = \text{MAX}
 \end{aligned}$$

El pasar la imagen de RGB a HSV nos permite hacer una discriminación entre lo que llamamos tonalidad de la imagen (HUE), saturación (SATURATION, cantidad de color puro o ausencia de blanco) y el valor del color o brillo (VALUE). Asimismo Matlab® conforma la imagen HSV como una matriz de dimensiones “640x480x3”: se obtienen tres matrices de 640x480 que en conjunto representen una sola imagen, cosa que no sucede con una imagen binaria o de niveles de gris, donde solamente se dispone de una matriz. El espacio HSV es interesante en Matlab® precisamente porque permite trabajar de forma independiente con una imagen “tono o tonalidad”, una imagen “saturación” o una imagen “brillo”, cuyas características propias permiten escoger una u otra base para nuestro algoritmo.

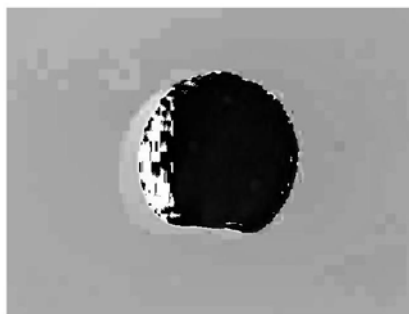


Figura 6 Imagen Hue de HSV de un melocotón

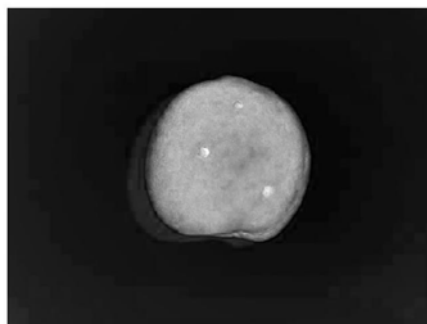


Figura 7 Imagen Saturación de HSV de un melocotón.

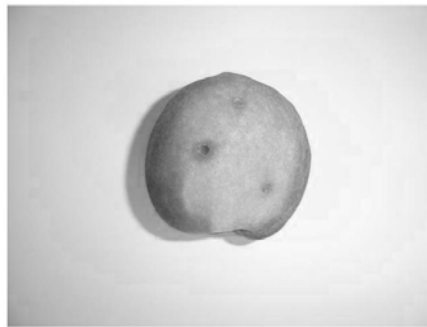


Figura 8 Imagen Value de HSV de un melocotón

Del resultado de transformar la imagen a HSV se decide que para la segmentación se usará o bien H o S, ya que la tercera y última matriz V, equivale a la matriz única resultante de haber pasado una imagen RGB a niveles de gris.

4.1.2 Thresholding. Determinación de umbrales

De los métodos expuestos para la segmentación anteriormente se ha escogido la segmentación por *thresholding* o uso de umbrales. La segmentación se consigue barriendo la imagen píxel a píxel i etiquetando cada píxel como “objeto” o “fondo”, según si el valor del píxel es mayor o inferior que un cierto valor L llamado umbral [2][Gonzalez, Woods,1993] aunque también es posible el uso de más de un umbral.

La selección de los umbrales es un aspecto fundamental en el *thresholding* ya que determina en gran medida la calidad del resultado final de la segmentación. Siendo así los umbrales no son escogidos al azar sino usando métodos tales como los histogramas.

La realización de histogramas permite en gran medida visualizar directamente la agrupación de los píxeles de una imagen mostrando gráficamente en dos

ejes por un lado los valores de los píxeles y por otro lado la frecuencia en que estos valores se producen en la imagen.

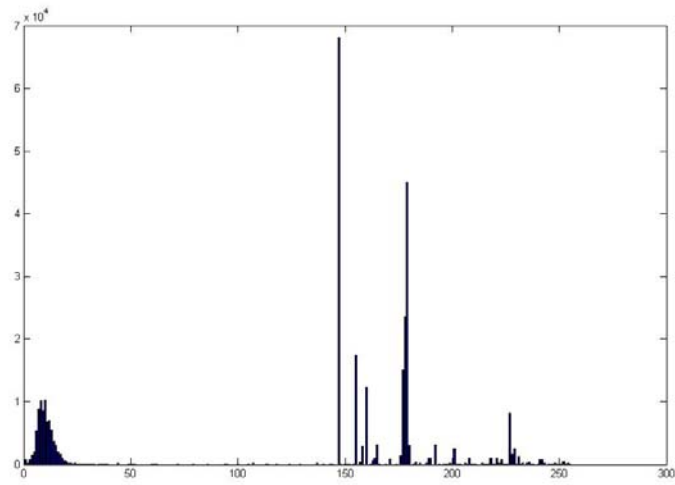


Figura 9 Histograma de la componente Hue de un melocotón sano.

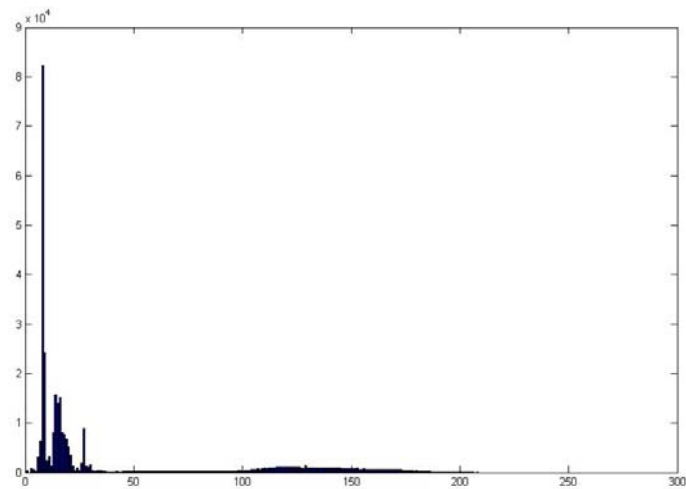


Figura 10 Histograma de la componente Saturación de un melocotón sano.

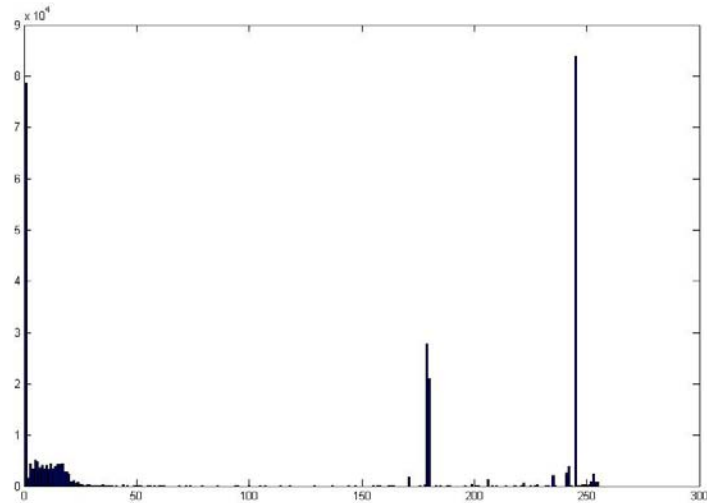


Figura 11 Histograma de la componente Hue de un melocotón defectuoso.

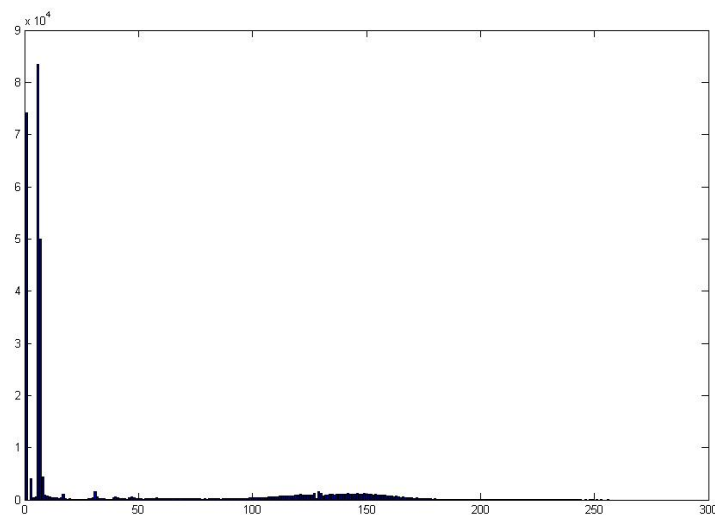
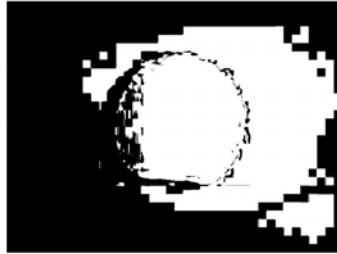


Figura 12 Histograma de la componente Saturación de un melocotón defectuoso.

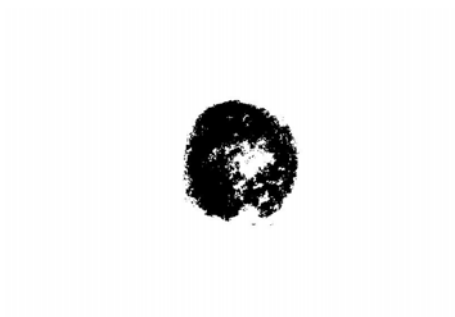
Observando varios histogramas como los de las figuras anteriores se decide binarizar con umbrales $L1=0$ y $L2=160$.

Los histogramas de H y de S son representados en el eje de ordenadas por Matlab® por valores de 0 a 255, aún siendo H y S matrices cuyo valor máximo de píxel es 1. Es necesario, pues, cambiar $L2=160$ por $L2=160/255$ (0,6227) para poder realizar la binarización o *thresholding*.

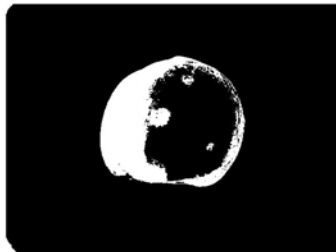
La función binarizar devolverá una imagen binaria a partir de la imagen H o S, L1 y L2 que son parámetros de entrada.



(a)



(b)



(c)

Figura 13 (a)Binarización de Hue con $L1=0$ y $L2=0,6227$ (b)Binarización de Saturación con $L1=0$ y $L2=0,6227$ (c)Binarización de Hue con $L1=0$ y $L2=0,6227$

Se aprecia claramente que la imagen S tiene invertido el objeto y el fondo.

4.1.3 Uso de operadores morfológicos. Apertura o cierre.

En principio sería lógico escoger la imagen H para la segmentación debido a que el melocotón aparece como blanco y el fondo permanece negro en su binarización. Aún así no podemos descartar la imagen S debido a los resultados que se expondrán a continuación.

Aunque los resultados de la binarización son aceptables es necesario eliminar el ruido de la imagen y rellenar huecos en el objeto a segmentar. Por ello se realizan pruebas de aperturas y cierres morfológicos con tres elementos estructurantes.

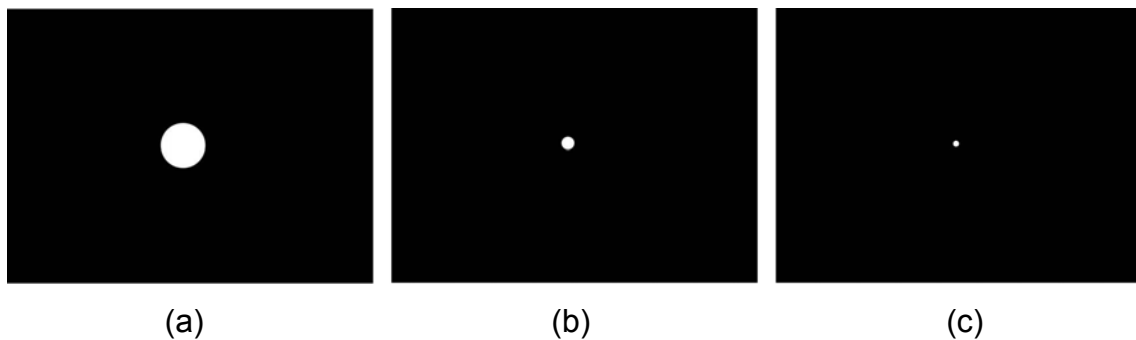


Figura 14 (a)Elemento estructurante “uno”. (b)Elemento estructurante “dos” (c)Elemento estructurante “tres”. Cuanto mayor el elemento estructurante más lento es el proceso de apertura y cierre morfológico de imágenes.

Estos tres elementos estructurantes son tres imágenes de fondo negro con un objeto circular de diferente tamaño. El elemento estructurante aplicado tiene esa forma ya que es la más afín a la forma de la pieza de fruta. Dicho círculo blanco es el elemento estructurante que barre la imagen y realiza las modificaciones según la operación morfológica asignada. Las pruebas han dado como resultado diferentes imágenes que se exponen a continuación.

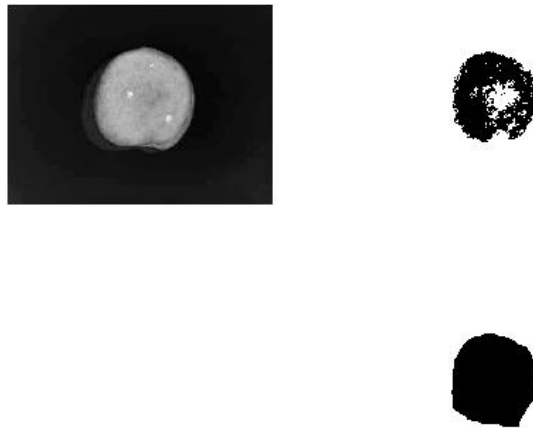


Figura 15 La imagen superior izquierda es la imagen de Saturación del espacio HSV. En la posición superior derecha se muestra la imagen de S binarizada con L1 y L2. En la parte inferior izquierda que aparece en blanco representa el cierre de S y al lado la apertura de la imagen Saturación con el elemento estructurante circular “uno”.

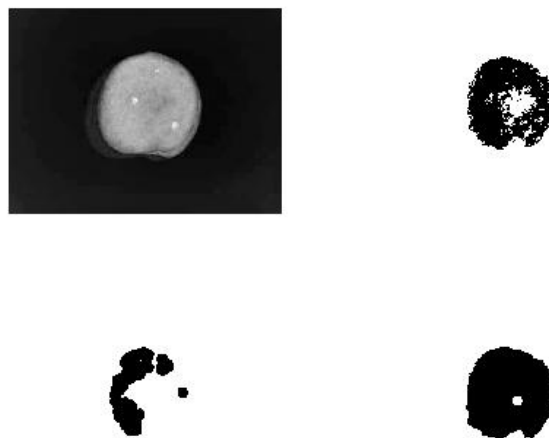


Figura 16 La imagen superior izquierda es la imagen de Saturación del espacio HSV. En la posición superior derecha se muestra la imagen de S binarizada con L1 y L2. En la parte inferior izquierda que aparece en blanco

representa el cierre de S donde toda la imagen se considera “objeto” y al lado la apertura de la imagen Saturación con el elemento estructurante circular “dos”. A diferencia de la figura 15 la imagen del cierre tiene parte de ella considerada como fondo.

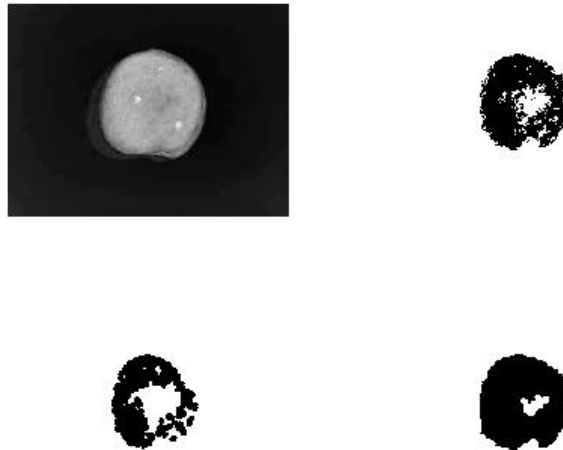


Figura 17 La imagen superior izquierda es la imagen de Saturación del espacio HSV. En la posición superior derecha se muestra la imagen de S binarizada con $L1$ y $L2$. En la parte inferior izquierda que aparece en blanco representa el cierre de S y al lado la apertura de la imagen Saturación con el elemento estructurante circular “tres”. Estas imágenes se consiguen con el menor de los elementos estructurantes.

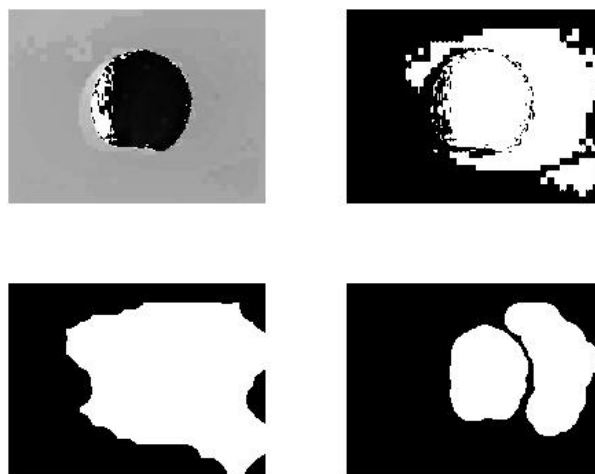


Figura 18 La imagen superior izquierda es la imagen de Saturación del espacio HSV. En la posición superior izquierda se muestra la imagen de S binarizada con L1 y L2. En la parte inferior izquierda que aparece en blanco representa el cierre de S y al lado la apertura de la imagen Saturación con el elemento estructurante circular “uno”.

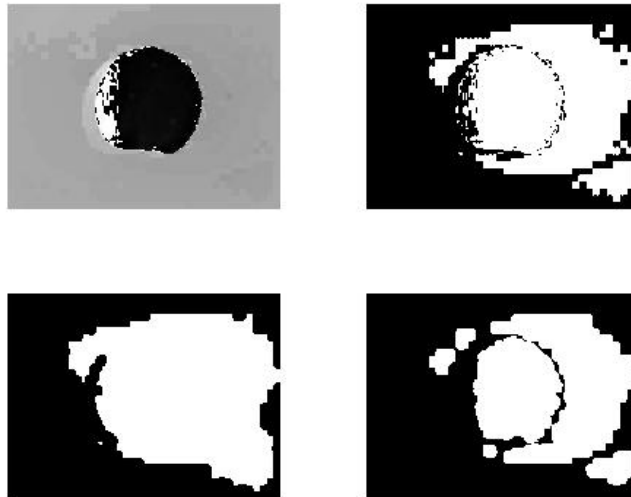


Figura 19 La imagen superior izquierda es la imagen Hue (de HSV). En la posición superior derecha se muestra la imagen H binarizada con L1 y L2. En la parte inferior izquierda representa el cierre de H y al lado la apertura de la imagen Hue con el elemento estructurante circular “dos”. Se observa que el resultado no es bueno para ninguna de las operaciones morfológicas.

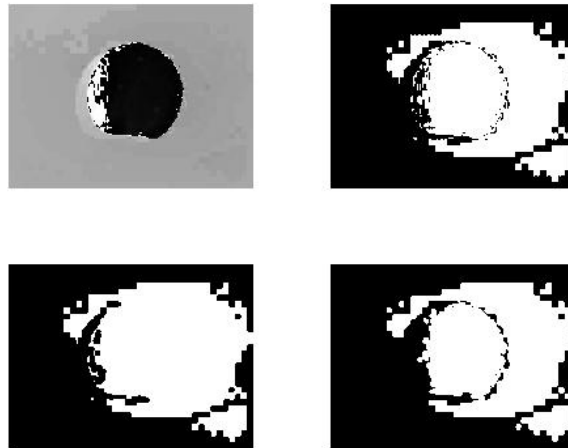


Figura 20 La imagen superior izquierda es la imagen Hue (de HSV). En la posición superior derecha se muestra la imagen H binarizada con L1 y L2. En la parte inferior izquierda representa el cierre de H y al lado la apertura de la imagen Hue con el elemento estructurante circular “tres”.

En este punto es necesario decidir qué imagen se toma para la segmentación y qué operación morfológica se aplica. Aunque a priori sea la imagen H de HSV la más apropiada por tener fondo negro y objeto blanco, se puede comprobar que S también puede dar un buen resultado, siempre y cuando se inviertan los valores que muestra, ya que estos están al revés de lo establecido: “1” o blanco es todo aquello considerado como objeto de interés (el melocotón) y “0” o negro el fondo.

Finalmente se ha escogido la imagen de Saturación (S) del espacio HSV al que se había convertido la imagen original. Después de binarizarla con los L1 y L2 escogidos anteriormente (el resultado es una imagen con dos únicos valores) se aplica una apertura morfológica (apréciease que el cierre de S no es de utilidad para la segmentación). Esta operación morfológica equivale a la aplicación de una erosión y una posterior dilatación a la imagen con el mismo elemento estructurante y se realiza con el objetivo de eliminar aquellos objetos o zonas de la imagen más pequeñas que el elemento estructurante circular escogido. Las zonas eliminadas son aquellas mas pequeñas que el elemento estructurante y de colores claros (blanco en caso de imágenes binarias).

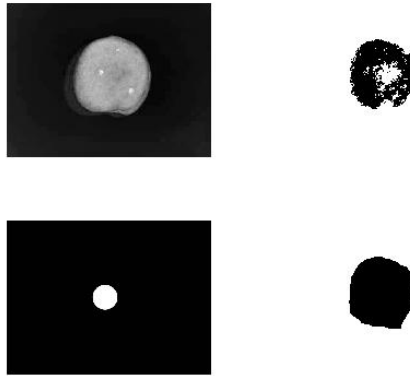


Figura 21 La imagen superior izquierda es la imagen Saturación (de HSV). En la posición superior derecha se muestra la imagen *S* binarizada con *L1* y *L2*. En la parte inferior izquierda se representa el elemento estructurante final “uno” y al lado la apertura de la imagen *S* binarizada.

Obsérvese en la figura 21 que el hecho de eliminar las zonas del melocotón más pequeñas que el elemento estructurante implica, para el caso de la imagen de la Saturación previamente binarizada, rellenar de negro lo considerado fondo: el melocotón. Siempre recordando que luego debe invertirse el resultado.

Para la parte del algoritmo destinada a la segmentación el elemento estructurante utilizado ha sido el número “uno”, de mayor tamaño que los demás. Esta característica ha permitido asegurar que éste solo pasara por las zonas convenientes al barrer la imagen (en el caso de *S* no se ha movido el elemento estructurante por el melocotón sino por el fondo, que en esta caso consta como elemento “objeto”).

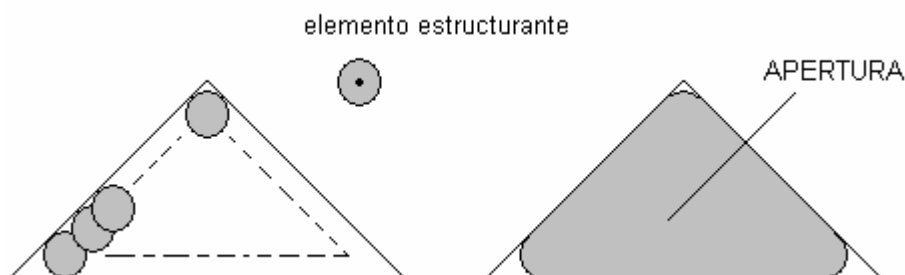


Figura 22 *Movimiento del elemento estructurante circular por una imagen de un triángulo. El resultado es otro triángulo de picos redondeados, pues el elemento estructurante no es capaz de llegar a las esquinas.*

La operación morfológica de cierre no era adecuada para el caso ya que aunque también suaviza el contorno del objeto como la apertura, a diferencia de esta suele fusionar roturas delgadas y zonas estrechas donde se percibe el fondo. También elimina pequeños agujeros y rellena huecos en el contorno del objeto, que es precisamente lo que ocurre cuando se aplica un cierre sobre la imagen S binarizada.

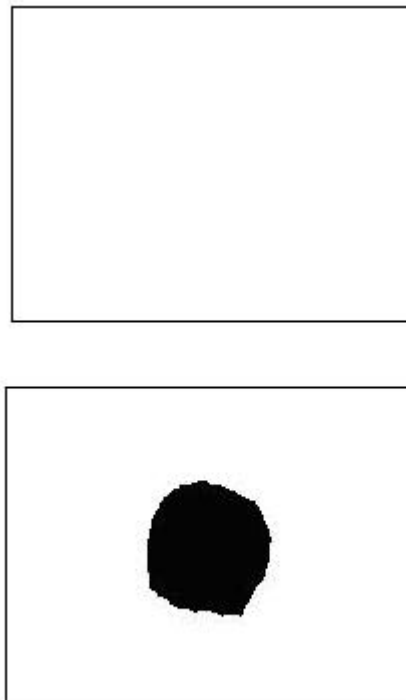


Figura 23 *Esta figura pretende ilustrar la gran diferencia entre el cierre morfológico y la apertura para el caso estudiado. La parte superior del marco está en blanco pues es la imagen que representa el cierre, mientras que la imagen de abajo contiene una mancha negra que es el resultado de la apertura a S binarizada. Obsérvese que las zonas blancas presentes en la zona del melocotón han sido eliminadas para dejar una mancha negra uniforme.*

En este punto de la segmentación es necesario invertir la imagen resultante del proceso anterior. Como muestra la figura 24 en la imagen binaria final queda como blanco la zona donde se sitúa el melocotón y negro el fondo.

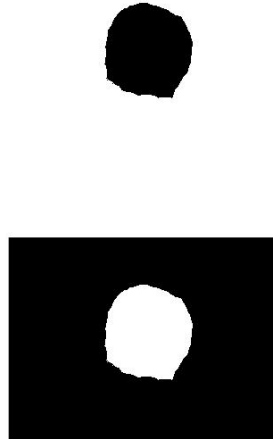


Figura 24 La imagen superior muestra la apertura de S binarizada mientras que la inferior muestra el resultado de invertir dicha apertura: la zona blanca es donde se sitúa el melocotón y donde se centra la segmentación.

Para realizar finalmente la segmentación es necesario sustituir los píxeles blancos por los valores de gris de la imagen original. También pueden sustituirse dichos píxeles por cada nivel de componente R, G y B, dando como resultado tres matrices que al representarlas unidas conforman una imagen segmentada en color.



Figura 25 La primera imagen muestra el resultado de segmentar la imagen sustituyendo los píxeles por los correspondientes niveles de gris. La imagen del medio realiza el mismo proceso pero sustituyendo los píxeles por las componentes RGB. Se crean tres imágenes (R, G y B) que mostradas juntas componen la imagen a color. La última imagen es la original de partida antes de la segmentación.

Si bien el perfil del melocotón no es perfecto sí es correcto para simplificar la determinación de la calidad de la fruta. Es importante que en la segmentación los defectos no hayan sido eliminados y considerados parte del fondo.

Obsérvese en la figura 25 que la sombra de la imagen original no ha sido incluida en la segmentación. Este detalle es muy importante para la detección de contornos, donde usaremos la imagen segmentada en niveles de gris.

4.2 Detección de contornos

El objetivo de este apartado es encontrar las características necesarias para poder establecer una discriminación entre melocotones sanos y melocotones defectuosos (esquema de la figura 4). Este paso se determina a partir de un cierto valor que relaciona los píxeles totales del melocotón con aquellos que pertenecen a un defecto.

Cuando el melocotón ha sido segmentado del resto de la imagen el siguiente paso se basa en la búsqueda de defectos o imperfecciones en la pieza. Éstos se caracterizan por una coloración y textura diferenciales dentro de la homogeneidad del melocotón.

Como se ha comentado en apartados anteriores, este estudio basa la toma de decisiones a base de pruebas sobre la imagen e intentando buscar un elemento común que sea válido para llegar al resultado deseado.

La detección de contornos es una opción que se contempló desde un principio. Para aplicar esta técnica se realizan una serie de pruebas con diferentes filtros de gradiente proporcionados por el Toolbox de Matlab® sobre imágenes en niveles de gris. Los filtros aplicados han sido: Sobel, Prewitt, Roberts y Canny, siendo finalmente el primero por el que se ha optado. Se decide Sobel porque a parte de detectar contornos, sus matrices (que actúan como una máscara sobre la imagen) tienen la ventaja de proporcionar un efecto diferenciador y suavizante. Debido a que este tipo de filtro también realza el ruido en la imagen, el efecto suavizador es una característica particularmente atractiva de los operadores Sobel.

$h_1[m,n]$			$h_2[m,n]$			(a)
-1	0	1	-1	-2	-1	
-2	0	2	0	0	0	
-1	0	1	1	2	1	

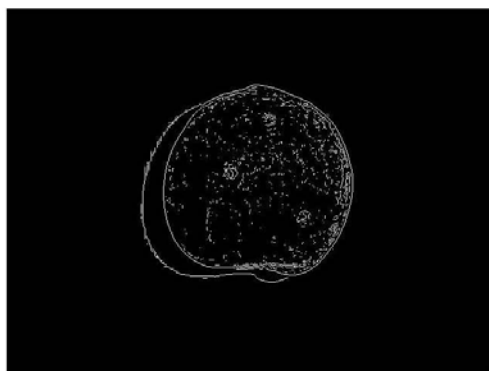
$h_1[m,n]$			$h_2[m,n]$		
-1	0	1	-1	-1	-1
-1	0	1	0	0	0
-1	0	1	1	1	1

(b)

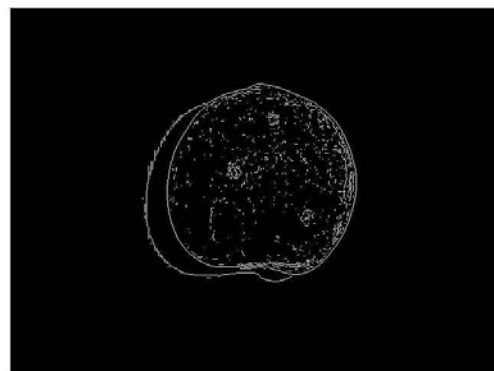
$h_1[m,n]$		$h_2[m,n]$	
1	0	0	1
0	-1	-1	0

(c)

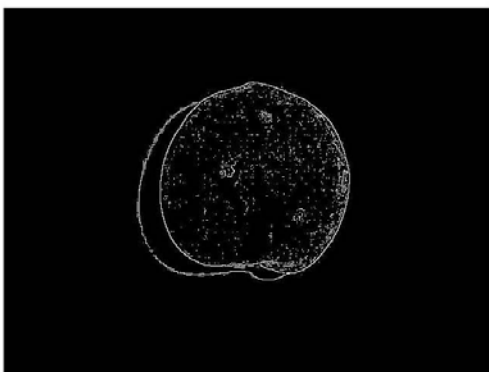
Figura 26 Los filtros derivadores o de gradiente horizontal –vertical (a) Filtro Sobel; (b) Filtro Prewitt; (c) Filtro Roberts. Son filtros que detectan cambios bruscos de intensidad en una determinada dirección, es decir bordes, que corresponden a altas frecuencias.



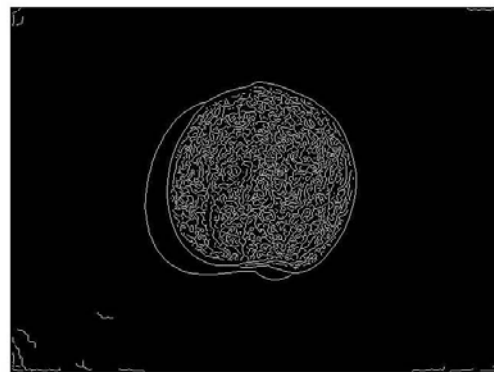
(a)



(b)



(c)



(d)

Figura 27 Resultado de aplicar los filtros de contorno o gradiente. (a)Sobel (b)Prewitt (c)Roberts (d)Canny. En los tres primeros puede apreciarse como ciertos píxeles se agrupan en el interior formando círculos pequeños que marcan tres agujeros formados por picotazos.

Canny ha sido eliminado del estudio directamente pues es demasiado certero en la detección de contornos y detecta incluso las texturas más insignificantes de un melocotón sano.

Aun habiendo descartado Canny el problema de detección excesiva de contornos persiste en el filtro Sobel, por lo que es necesario aplicar otro tipo de filtro que suavice el melocotón lo suficiente como para no detectar cambios de tonalidad (como si fueran defectos) en los melocotones sanos. Aun así el suavizado debe permitir hallar texturas allí donde sí hay imperfecciones.

Para solucionar el problema se aplica, antes de la detección de contornos con un filtro Sobel, un filtro promediador o de mediana. Este filtro suavizará el melocotón para que el detector de contornos no tenga en cuenta los cambios de intensidad de la propia piel del fruto. El tamaño del filtro de promedio será tal que permita obtener contornos después del filtro Sobel.

1	1
1	1

 $\times \frac{1}{4} \quad (a)$

1	1	1
1	1	1
1	1	1

 $\times \frac{1}{9} \quad (b)$

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

 $\times \frac{1}{16} \quad (c)$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

X1/25 (d)

Figura 28 (a)(b)(c)(d) Los filtros de promedio o mediana estudiados

El filtro promediador se aplica directamente sobre la imagen en niveles de gris, igual que el filtro de contornos Sobel.

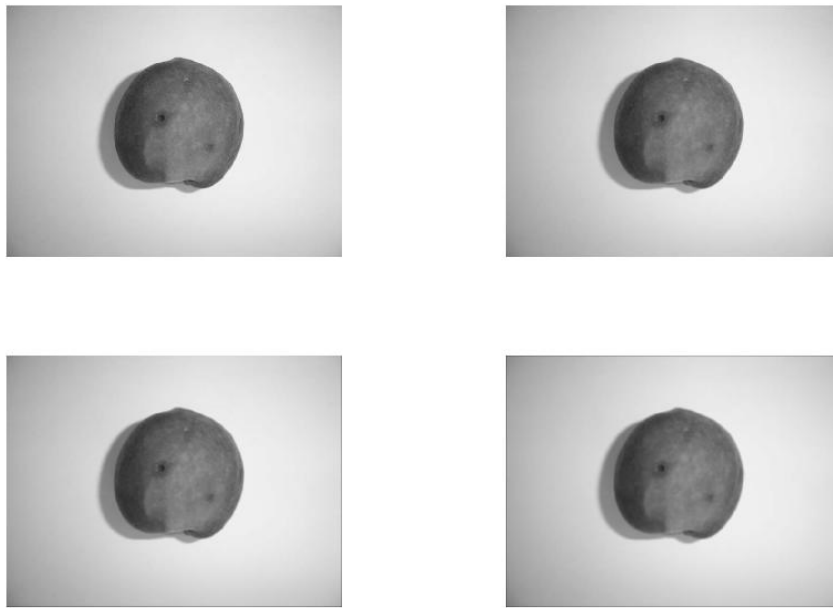


Figura 29 Las siguientes figuras muestran el resultado de aplicar un filtro de promedio a una imagen. Para la imagen superior izquierda se ha usado el filtro (a) de la figura 28. Para la imagen superior derecha se ha usado el filtro (b) y para las dos siguientes el (c) y el (d) respectivamente. En todos ellos se aprecia un leve suavizado.

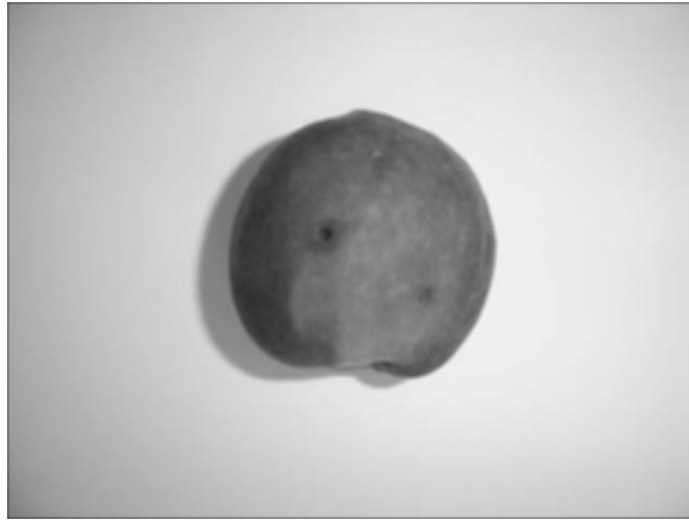


Figura 30 Esta imagen muestra el resultado de aplicar el filtro de tamaño 5x5, finalmente el escogido para el proyecto.

Los efectos de aplicar un filtro de promedio previo al de contornos se puede apreciar en las siguientes imágenes:

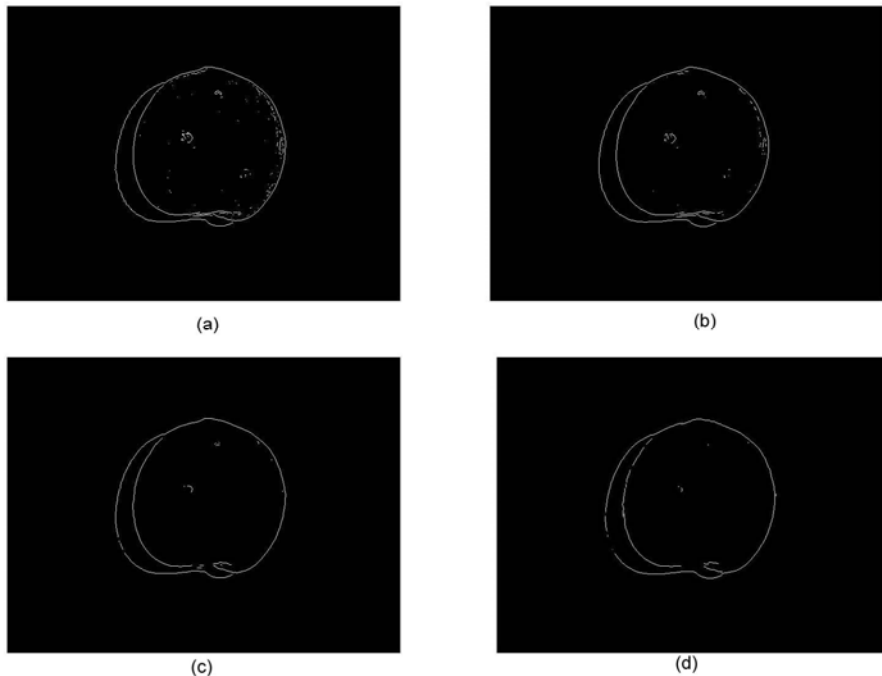


Figura 31 Las siguientes imágenes muestran resultado de aplicar un filtro de contornos Sobel según se ha aplicado antes (a) el filtro promediador de 2x2, (b) 3x3, (c) 4x4 o (d) 5x5.

El filtro Sobel de Matlab® recibe una imagen en niveles de gris y devuelve otra binaria, marcando como negro el fondo y como blanco todos los contornos hallados. Si bien es cierto que el resultado es bueno, también es posible incluir en el filtro Sobel un parámetro de entrada diferente de la propia imagen a filtrar: un threshold, que para el estudio se fija en 0,03.

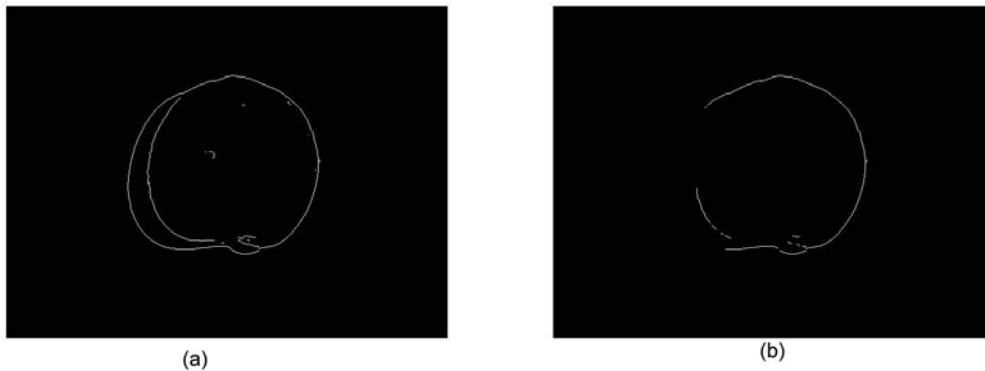


Figura 32 (a) Filtro Sobel con tres a 0,03 y (b) Filtro Sobel a 0.05.

Cuando se hallan los bordes se debe hallar alguna relación que permita clasificar los melocotones en sanos y defectuoso. Tal y como se ha introducido al principio de este apartado, se toma como valor discriminatorio la relación entre los píxeles totales de dentro del melocotón y los píxeles hallados defectuosos después del Sobel (marcados como blancos), también de dentro del melocotón. Es importante destacar que los píxeles de la relación deben referirse exclusivamente a la parte de la superficie del fruto, excluyendo así el contorno de la sombra presente en la salida del Sobel y el contorno propio del melocotón.

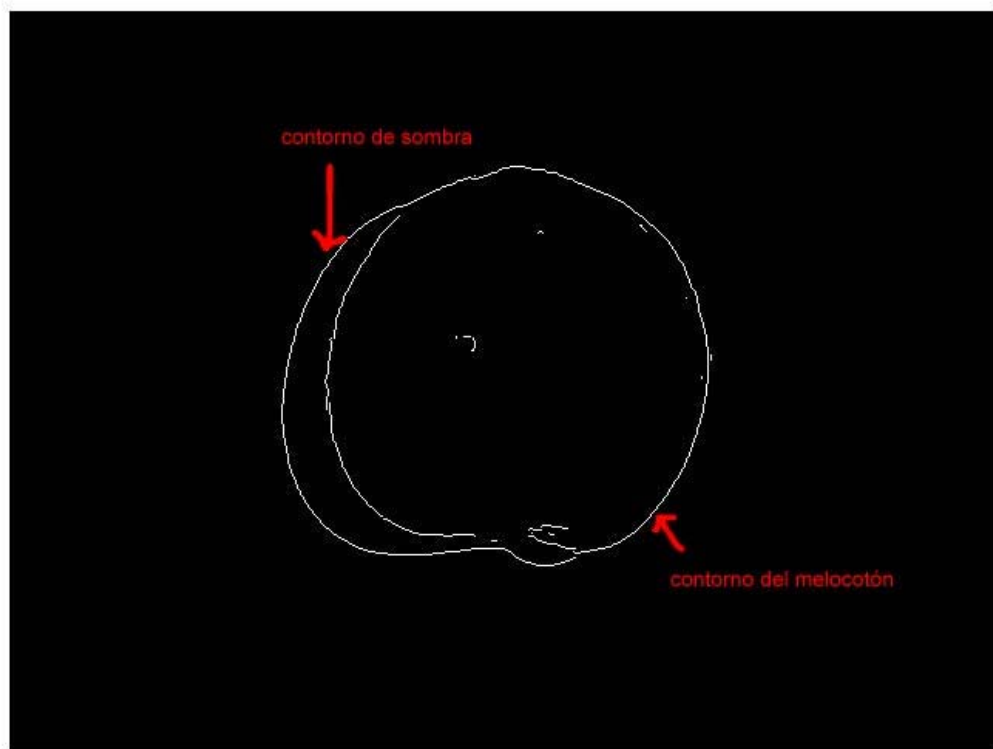


Figura 33 La imagen muestra los píxeles a excluir en la relación “valor”: son todos aquellos que pertenecen a la sombra y al perfil del fruto.

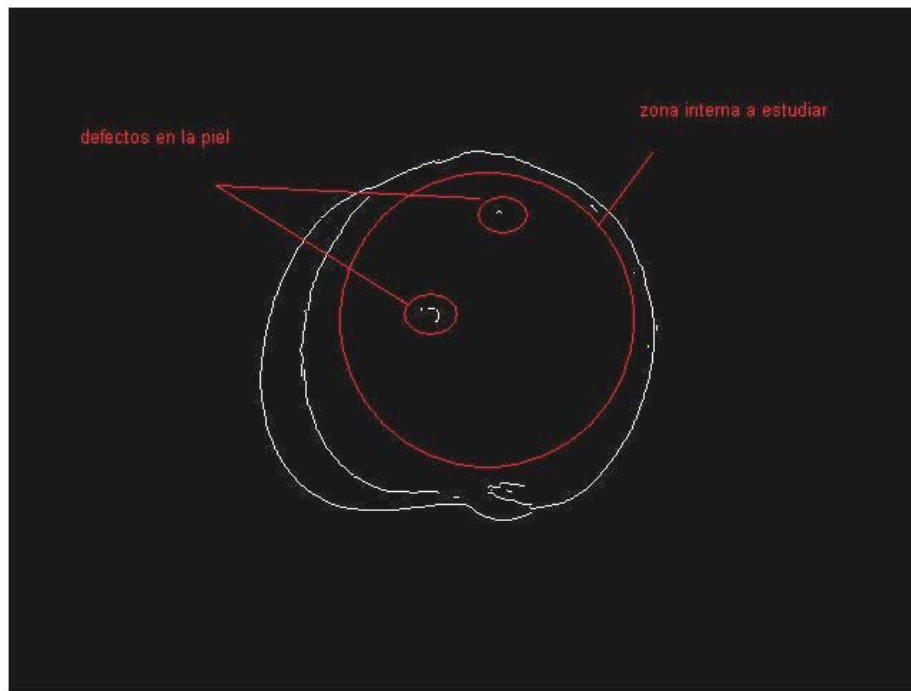


Figura 34 La imagen de la figura muestra que la zona a contar píxeles es la interior y que es suficiente con que haya algunos píxeles que muestren los defectos, no siendo necesario que se aprecie el perfil perfecto del defecto.

El valor será 0 si NO tenemos píxeles de contornos de dentro del melocotón en la imagen filtrada por el Sobel, mientras que será distinta de cero para aquellas frutas en las que sí se detecten imperfecciones dentro del melocotón.

$$\text{valor} = \frac{\text{píxeles blancos dentro del melocotón (a)}}{\text{píxeles totales dentro del melocotón (b)}}$$

Figura 35 Relación entre píxeles del melocotón. Es necesario dividir los píxeles blancos que son aquellos que indican contornos de defecto entre los píxeles totales de dentro del melocotón. Así se normaliza el tamaño del melocotón teniendo en cuenta únicamente su figura y dejando fuera de estudio los píxeles de contorno y de sombra que no forman parte del fruto y son innecesarios para el estudio. Esta proporción indica que si en algún momento la relación da 0 es

porque el numerador es 0 y no existen píxeles defectuosos dentro de la zona del melocotón.

Valor = 0 → número de píxeles blancos dentro del melocotón=0

Valor ≠ 0 → número de píxeles blancos dentro del melocotón≠0

Para el valor = 0 se considera el melocotón como sano o golpeado, mientras que para el valor ≠ 0 se considera el melocotón como defectuoso, tal como muestra el esquema del sistema en la figura 4.

Para contar el número de píxeles dentro del melocotón se ha creado una función que recorre la imagen de contornos y la compara con la imagen segmentada. Como se destacó en el apartado de "segmentación", la imagen resultante de la segmentación no tiene sombra, puesto que ha sido eliminada durante el proceso de binarización. El algoritmo usado para contar (a) y (b) (obsérvese la figura 35) recorre píxel a píxel las imágenes de entrada –imagen segmentada e imagen de contornos- y devuelve una imagen de contornos internos del melocotón (excluye el perfil del melocotón y la sombra) y el número "valor".

El recuento de píxeles se hace en base al siguiente criterio:

1. Si el píxel vale 0 (negro) en la imagen de contornos y 0 (negro) en la segmentada, significa que el píxel pertenece al fondo y no interesa contarlos. Se da el valor de 0 a ese píxel en la imagen que devuelve la función.
2. Si el píxel vale 1 (blanco) en la imagen de contornos y 0 (negro) en la segmentada, significa que el píxel pertenece al contorno de sombra de la imagen de contornos y no interesa contarlos. Se da el valor de 0 a ese píxel en la imagen que devuelve la función.

3. Si el píxel vale 1 (blanco) en la imagen de contornos y cualquier otro valor en la segmentada, significa que el píxel es de una zona defectuosa del interior del melocotón y debe incrementarse el valor de (a). También se incrementa el valor de (b) y se da el valor 1 al píxel que corresponde en la imagen de salida.

4. Si el píxel vale 0 (negro) en la imagen de contornos y cualquier otro valor en la imagen segmentada, significa que el píxel pertenece al interior del melocotón aunque no se trata de un defecto. En este caso solo se incrementa el denominador (b) y se le da al píxel en cuestión de la imagen de salida el valor 0.

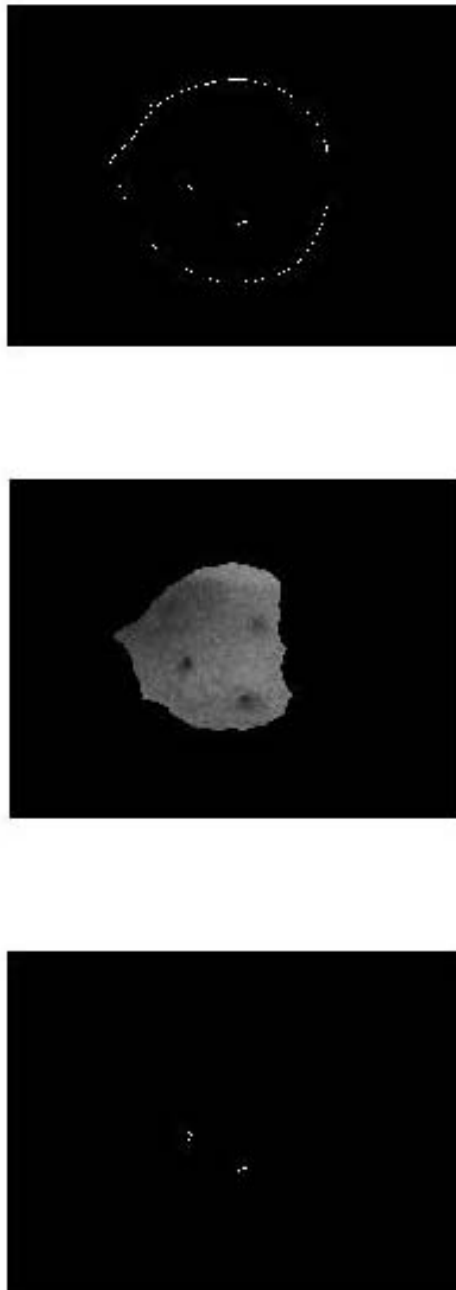


Figura 36 La imagen muestra la imagen de contornos del filtro Sobel, que junto a la imagen segmentada de la posición del medio se obtiene la imagen de contornos interiores del melocotón (la imagen de la última posición). En este caso la imagen de salida tiene algunos contornos blancos, que se corresponden a los agujeros de la imagen segmentada.

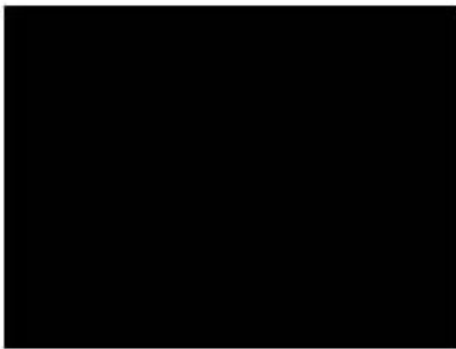
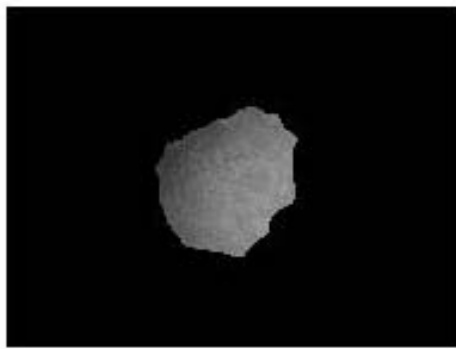
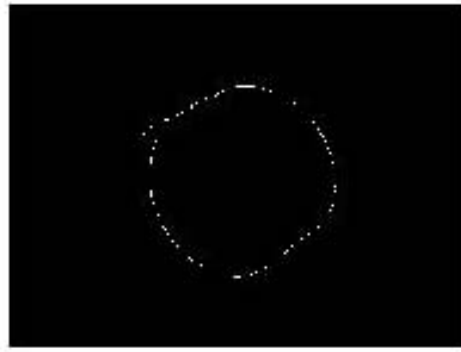


Figura 37 La imagen muestra la imagen de contornos del filtro Sobel, que junto a la imagen segmentada de la posición del medio se obtiene la imagen de contornos interiores del melocotón (la imagen de la última posición). En este caso la imagen de salida no tiene ningún contorno blanco, ya que el melocotón es sano. Para esta situación el parámetro “valor” será igual a 0.

En este punto del esquema la línea a seguir se separa en dos: una lleva a determinar si el melocotón es sano o no y la otra a la clasificación directa de defectos.

4.3 Cómputo de contenedores en histogramas.

En este bloque representado en el esquema 4 entran todos aquellos melocotones que tienen “valor = 0” en la proporción de la figura 35.

En este punto el problema surge al comprobar que tanto los melocotones considerados sanos como los golpeados pasan por el mismo bloque, como si ambos fueran válidos. Esto se debe a que los melocotones golpeados no tienen roces, texturas ni desgarros en la superficie de la piel, por lo que no son detectados en el filtro de contornos Sobel, pero sí se aprecia en ellos una coloración diferente: gran parte de la superficie muestra una tonalidad aproximadamente marrón o negruzca bastante uniforme que caracteriza los golpes en este tipo de frutos.

Los melocotones al madurar frecuentemente caen del árbol antes de ser recolectados por el agricultor. El golpe de la caída más el calor del sol y la humedad del suelo de los árboles favorece la aparición de moratones en el melocotón. También se producen golpes al ser recogidos con cubos y luego ser vertidos en cajas mayores que van hacia los centros logísticos donde se realizará la criba. Los golpes en la fruta tienen la característica que al producirse inicialmente no son detectados hasta al cabo de unas horas cuando la coloración marrón empieza a aflorar. Esto implica que un melocotón que parecía sano al recolectarlo puede al cabo de unas horas no serlo al clasificarlo.



Figura 38: ejemplos de melocotones golpeados. Se aprecia en ellos una zona oscura bastante uniforme en color si es comparada con la zona sana.



Figura 39 Ejemplos de melocotones sanos fotografiados con diferente iluminación. Obsérvese la variedad de tonos: amarillos, naranjas y rojizos.

Si se representa el histograma en niveles de gris de un melocotón sano y uno golpeado puede apreciarse la diferencia en la distribución de los píxeles. Si bien es cierto que en estos histogramas parte de los píxeles hacen referencia al fondo, es posible descartarlos usando el histograma de la imagen segmentada en gris.

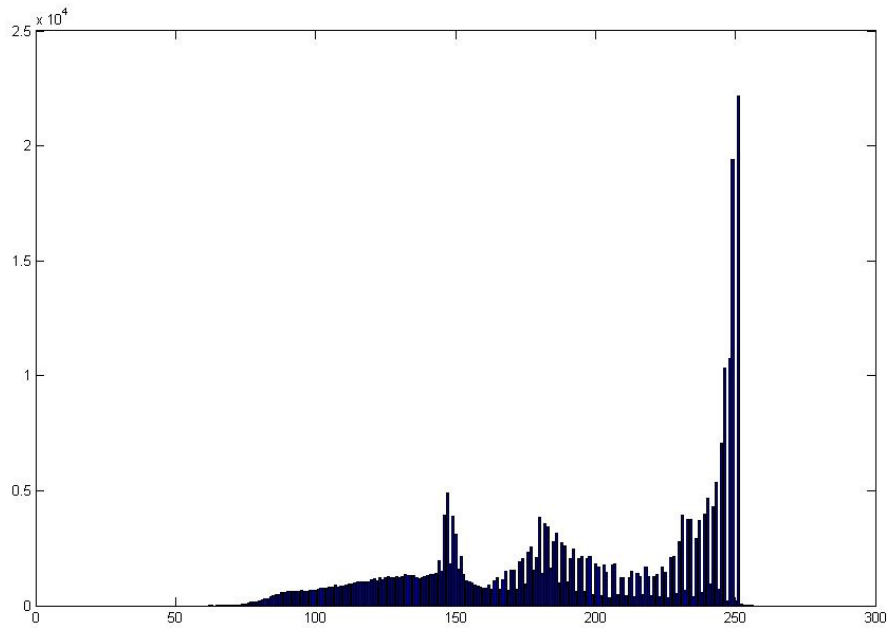


Figura 40 Histograma de la imagen de un melocotón sano en niveles de gris

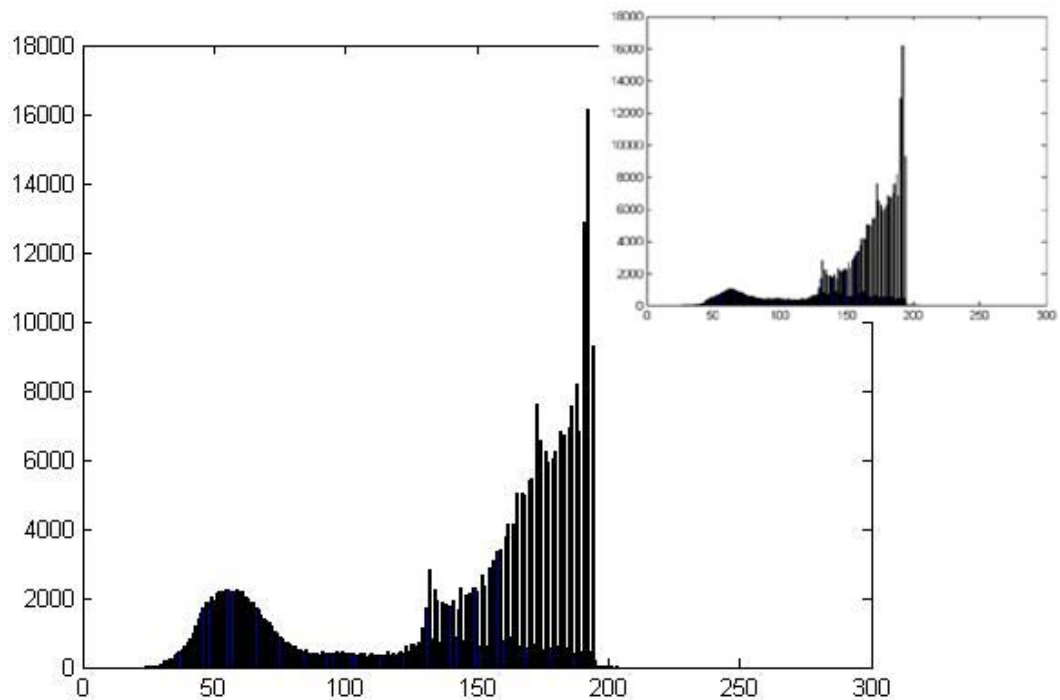


Figura 41 Histograma de una imagen en niveles de gris de un melocotón golpeado. En pequeño se encuentra el histograma del fondo de la imagen. Se puede apreciar la diferencia entre ambos histogramas por la forma que tienen de acumularse los píxeles. La diferencia entre uno y otro da como resultado la zona del melocotón golpeado. Obsérvese que los píxeles del fruto son los que

se acumulan a la izquierda en el grupo entorno al nivel 50. Los píxeles más a la derecha son más claros y pertenecen al fondo, tal como se aprecia en la imagen del margen superior derecho. En principio es complicado discernir de este histograma de melocotón golpeado del sano de la figura anterior.

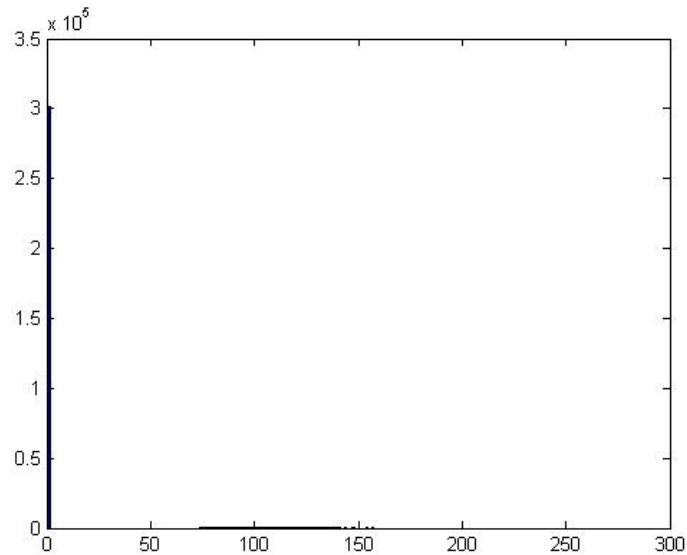


Figura 42 Histograma de la imagen segmentada de un melocotón golpeado. En el valor 0 se acumulan los píxeles pertenecientes al fondo negro de la segmentación.

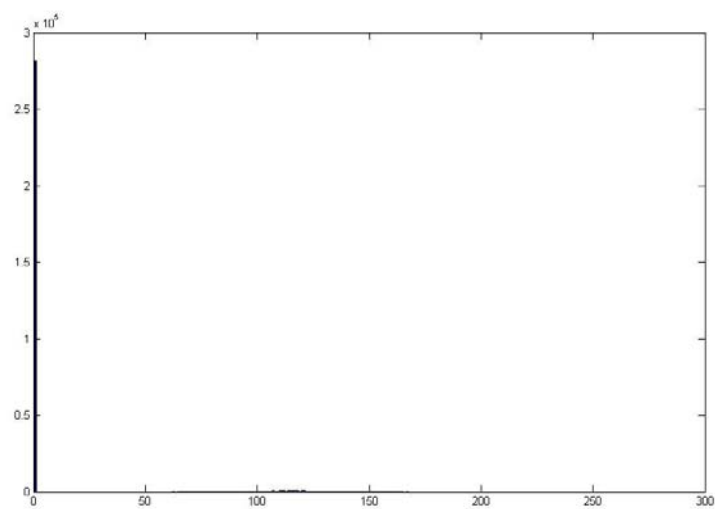


Figura 43 Histograma de la imagen segmentada de un melocotón sano. Se aprecia claramente un máximo para la posición 0 (negro) del histograma: son todos los píxeles pertenecientes al fondo.

En el histograma de la imagen segmentada en niveles de gris existe un máximo en el valor 0 que representa el conjunto de píxeles que conforman el fondo de la imagen. Si se pone a cero el primer contenedor (*bin*) del vector y representamos de nuevo los histogramas se aprecia que esta vez todos los píxeles son de fruta.

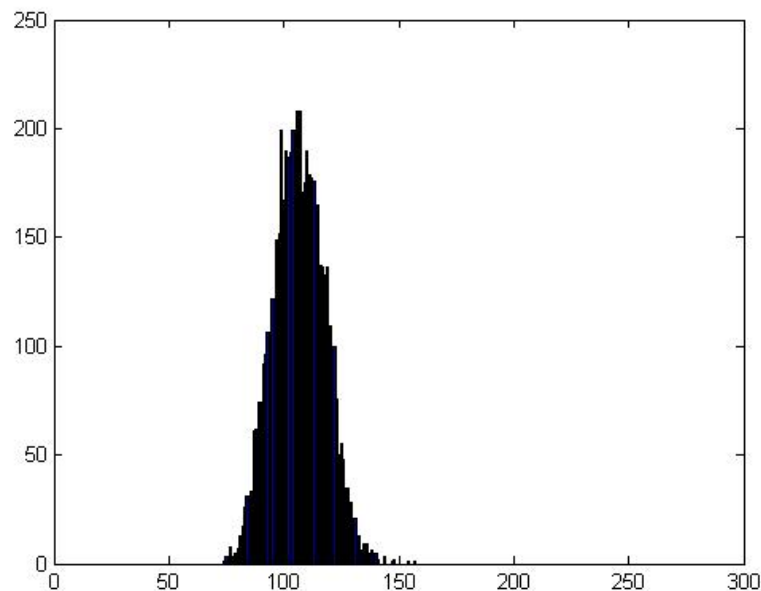


Figura 44 Histograma de un melocotón golpeado, poniendo a cero los píxeles de fondo que ocupaban la posición cero del vector histograma. Inicialmente no parece que exista ninguna diferencia significativa entre el histograma del melocotón golpeado y el sano de la figura siguiente. Sin embargo, es interesante observar el número de contenedores o valores que abarca este gráfico de la figura en comparación con el del fruto sano, obsérvese la anchura.

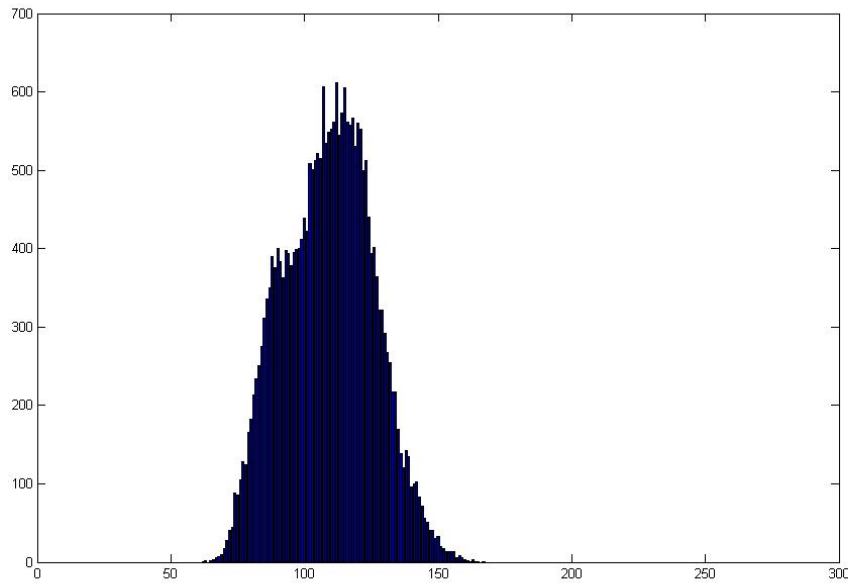


Figura 45 Histograma de un melocotón sano, poniendo a cero los píxeles de fondo que ocupaban la posición cero del vector histograma.

Se observa que la distribución de píxeles es diferente de un melocotón sano a un melocotón golpeado. Un melocotón sano se compone por tonos amarillos, rojos, naranjas y sus transiciones mientras que un melocotón golpeado tiene una gran mancha homogénea de color oscuro. Aunque inicialmente esto parece no quedar reflejado en los histogramas, si es cierto que el número de contenedores o píxeles en el eje “x” es diferente en uno y otro caso. El motivo posiblemente se encuentra en los tonos que un melocotón sano tiene, como se ha mencionado van del amarillo al rojo, pasando por tonos anaranjados e incluso alguna pincelada de marrón, más oscura. Estos tonos al convertirse a niveles de gris ocupan un ancho de valores mayor que no en el caso de frutos golpeados, cuyas tonalidades no son tan ricas en variación. Un melocotón golpeado suele tener tonos rojizos, marrones, poco de naranja y casi nada de amarillo. Estos tonos últimos son más claros al pasarse a niveles de gris y son contenedores (posiciones del vector histograma) desocupados. Esta variación de tono es debida a que los melocotones golpeados suelen madurar a más velocidad perdiendo los tonos amarillos del melocotón que está madurando, pero aun está duro (no verde, sino este tono también se reflejaría).

Según el razonamiento anterior se puede suponer que el número de contenedores (número de casillas o posiciones) ocupados del vector histograma será menor para el caso de melocotones golpeados. Bajo este supuesto se realizan tablas comparativas entre melocotones sanos y golpeados calculando el número de posiciones del vector diferentes de cero, es decir ocupadas. Este estudio da como resultado que si el histograma es de un melocotón golpeado, el número de contenedores ocupados es aproximadamente inferior al 29-30% del total (se entiende total por 256, ya que el histograma tiene posiciones de 0 al 255). Para números superiores el melocotón se considera sano.

El bloque denominado “Cómputo de contenedores en histogramas”, simplificando, determina si un melocotón es sano o está golpeado en función de el número de contenedores ocupados en el histograma. Si el número es inferior a 75 (aproximadamente un 29-30% del total del histograma) se considera que el melocotón defectuoso con una imperfección llamada “golpe”.

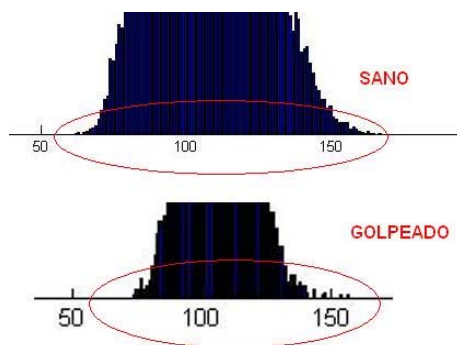


Figura 46 Esta imagen ilustra el ancho del histograma o número de contenedores ocupados. Generalmente dicho número es mayor para melocotones sanos.

4.4 Análisis de la mediana en HSV

El esquema del sistema representado en la figura 4 muestra que este último bloque da como resultado la clasificación del melocotón según dos tipos de defectos: picotazos y roces que dejan la pulpa visible.

Una vez la relación valor da una cifra distinta de 0 ya significa que el melocotón no es válido para su consumo directo y que será procesado para darle otro uso: licuarlo para la producción de zumos o triturarlo para el envasado de mermeladas.

Sin embargo es posible que se necesite saber de qué tipo de defecto se trata para poder realizar algún tipo de estudio sobre plagas, consecuencia de aplicaron de productos fitosanitarios, daños causados por temporales de nieve, granizo... Todas estas situaciones dejan marcas en la fruta que pueden ser de algún modo útiles.

Para este proyecto se clasifica el tipo de daño en la superficie del melocotón en función de si se trata de un golpe, de una marca producida por un picotazo de ave o por un desgarro en la piel que deja ver la pulpa amarilla (para la variedad estudiada).



(a)



(b)



(c)



(d)

Figura 47 Melocotones con defectos. (a) es defecto de picotazos mientras que (b), (c) y (d) son defectos de desgarro en la piel y dejan a la vista la pulpa del fruto.

Para clasificar el tipo de imperfección se estudia también la distribución de píxeles en el histograma de frutas con picotazos y con desgarros en la piel. Dichos histogramas se hacen a partir de la imagen S (Saturación) del espacio HSV al que ya se había transformado la imagen. Los picotazos tienen tonos anaranjados y marrones mientras que los desgarros dejan ver directamente tonos amarillos.

El histograma de S muestra la distribución de los píxeles según el nivel de intensidad de color o ausencia de blanco. Los colores más vivos tienen valores de saturación más elevados que aquellos más apagados, que parecen mezclados con blanco. Los rojos, amarillos y anaranjados del melocotón aparecen como tonos grisáceos, pero claros, en la imagen de saturación de HSV, mientras que el fondo claro queda negro en dicha imagen. Los defectos quedan resaltados en blanco y toman los valores máximos de saturación.

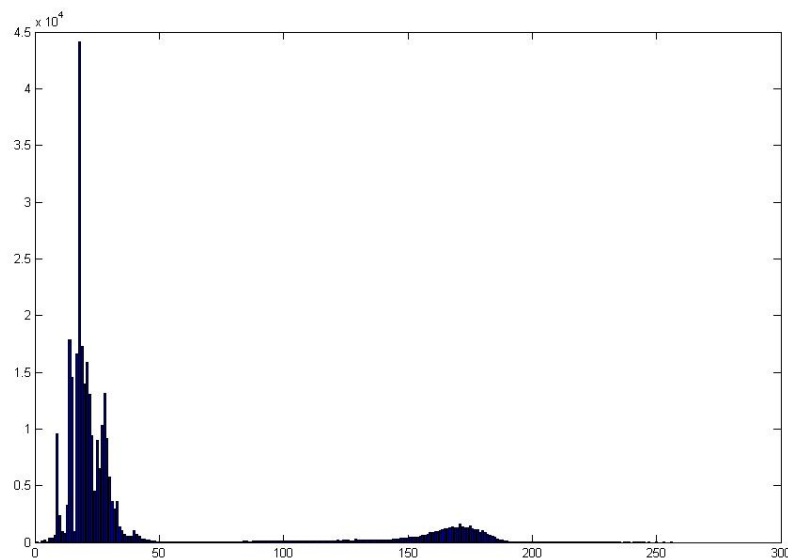


Figura 48 Histograma de S de un melocotón picoteado.

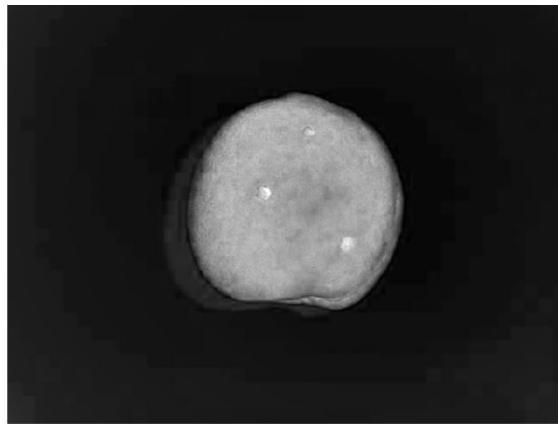


Figura 49 Imagen de Saturación de un melocotón picoteado. Es destacable la coloración de los agujeros del fruto.

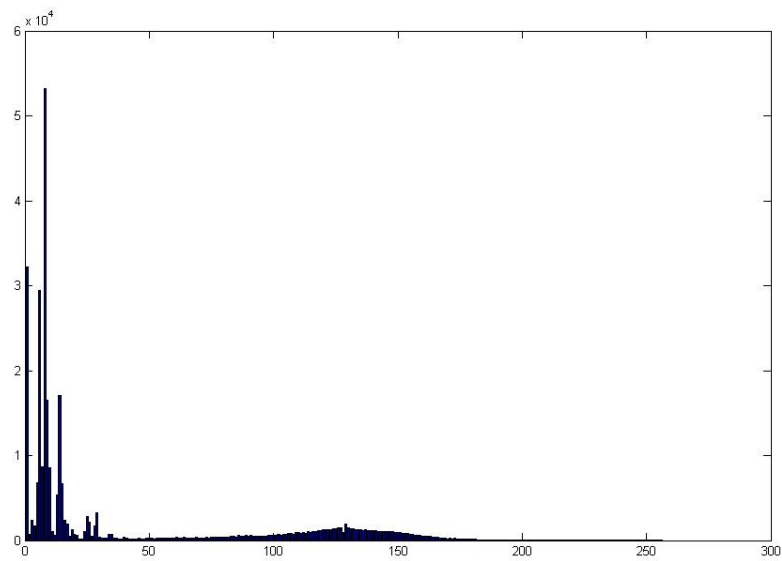


Figura 50 Histograma de S de un melocotón con desgarro en la piel.



Figura 51 *Imagen de Saturación de un melocotón con pulpa visible. Es destacable el tono blanco de los desgarros del fruto.*

Los histogramas son muy parecidos y muestran una agrupación similar de los píxeles, sobretudo en los valores próximos a cero, que corresponden al fondo de la imagen, no al melocotón. Como los defectos resultan ser una mínima parte del histograma, pues sólo representan un bajo porcentaje respecto la imagen S total, quedan agrupados en los valores más próximos al blanco (255).

Teniendo en cuenta que los píxeles blancos del defecto representan aquellas barras del histograma con menor número de píxeles, se centra la atención en este hecho y se aplica al histograma un cálculo de mediana.

La mediana estadística del histograma de S, si bien pierde la información de saturación, da un punto central en los valores de agrupaciones de píxeles. El cálculo de la mediana consiste en ordenar las columnas del histograma (cantidades de píxeles de iguales características) de menor a mayor y tomar de referencia el valor que hay en la posición central (el que se encuentra en la posición 127-128 de 256 contenedores). Organizándolo así el histograma los defectos que eran columnas pequeñas próximas al blanco quedan ahora posicionados en la parte izquierda del histograma.

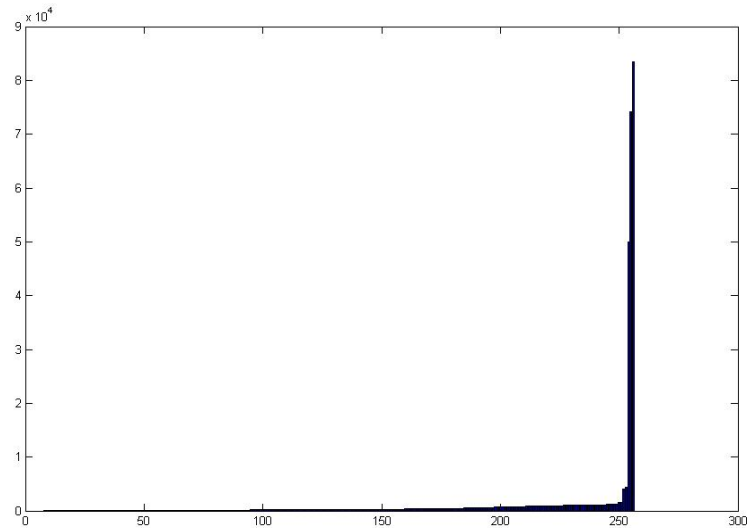


Figura 52 Histograma de la figura 48 ordenado por el método de la burbuja. Las columnas de píxeles se han redistribuido en orden creciente para este melocotón picoteado.

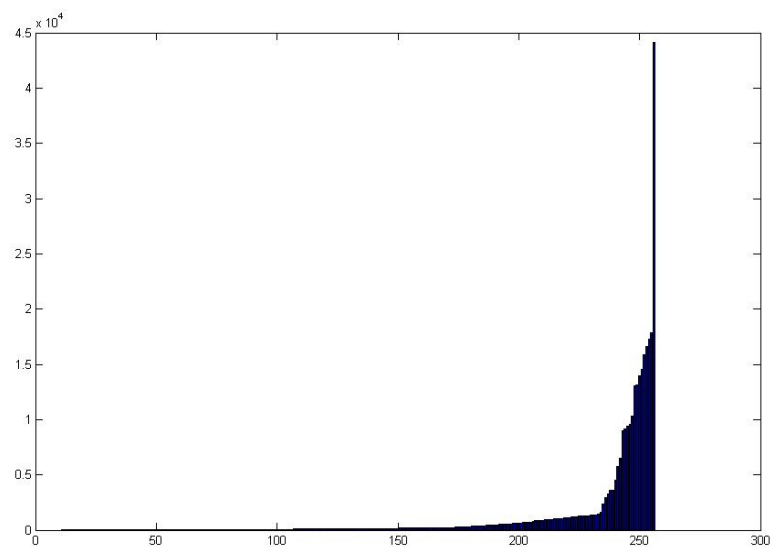


Figura 53 Histograma de la figura 50 ordenado por el método de la burbuja. Las columnas de píxeles se han redistribuido en orden creciente para este melocotón con pulpa visible. Se aprecia la diferencia entre los valores que alcanzan las columnas en un caso y otro.

El método de ordenación aplicado es el método de la burbuja, cuyo uso tiene como única función la representación del histograma ordenado, pues así es

posible marcar un punto de referencia con la mediana a la hora de discriminar entre melocotones picoteados y melocotones con pulpa visible.

Los defectos de melocotones con marcas de picos de ave o de granizo son representados por un número menor de píxeles que no el caso de melocotones con desgarros en la piel. Las columnas de píxeles de iguales características del histograma también tienen diferente longitud. Basando el uso de la mediana en eso es posible determinar un umbral a partir del cual si la mediana es superior o inferior clasifique uno u otro tipo de imperfección.

Se establece como umbral tras diversas pruebas un valor de mediana de 180. A partir de ese valor los melocotones con medianas en el histograma superiores a 180 serán aquellos que tienen pulpa visible y viceversa para los picoteados.

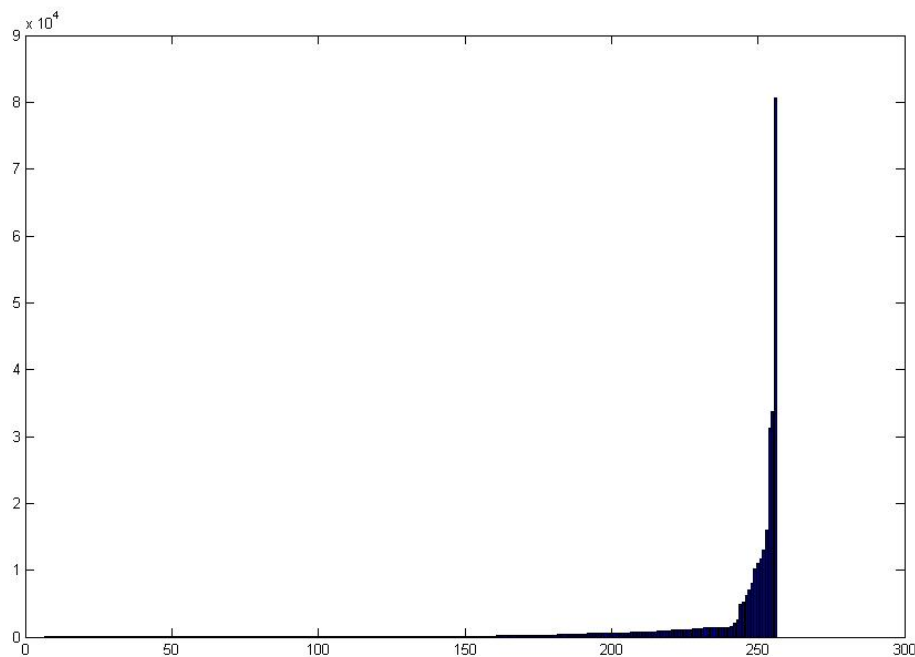


Figura 54 Histograma ordenado de la saturación de un melocotón con agujeros producidos por picotazos. Obsérvese que la columna central (posición 127-128) casi no se aprecia y no se produce un aumento del nivel hasta la posición 150 aproximadamente. Este sería el caso en que la mediana < 180.

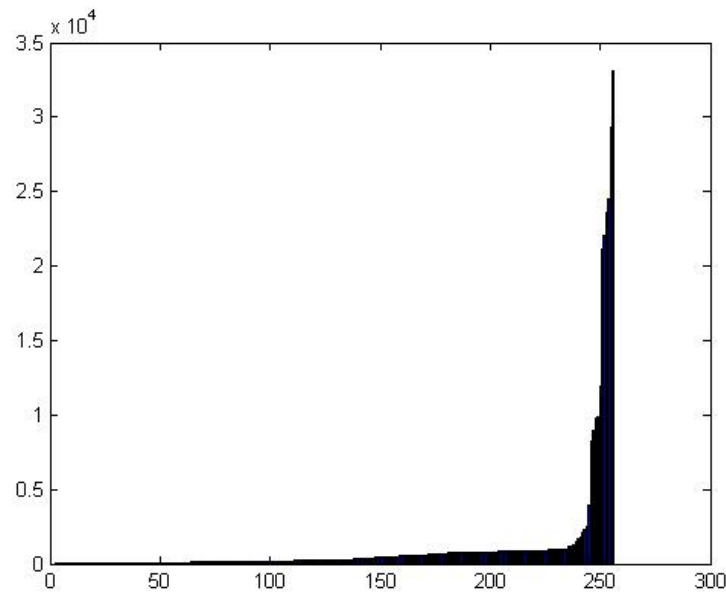


Figura 55 Histograma ordenado de la saturación de un melocotón con pulpa visible. Obsérvese que la columna central (posición 127-128) si tiene cierta altura a diferencia de la figura anterior. Este sería el caso en que la mediana > 180.

Para mostrar cuáles son los píxeles detectados como defectuosos y verificar que se corresponden a los defectos del melocotón se ha empleado una técnica de etiquetado de píxeles conexos y se ha creado una función que marca con cruces rojas las zonas de la imagen donde el selector ha detectado imperfecciones.

Esta función solo se ejecuta para melocotones cuyo parámetro “valor” es distinto de 0, ya que el etiquetado depende de los contornos hallados.

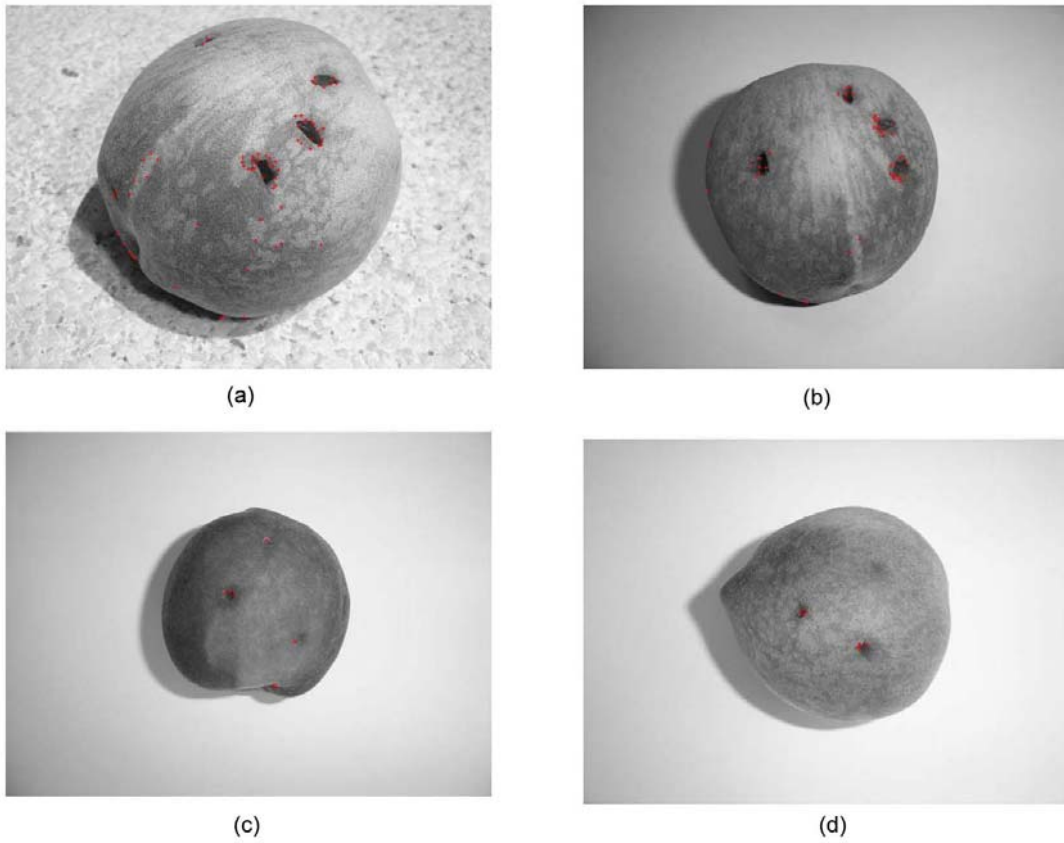


Figura 56 Las imágenes (a) (b) (c) y (d) muestran las zonas que el selector ha detectado como defectuosas marcándolas con cruces rojas en melocotones con picotazos.

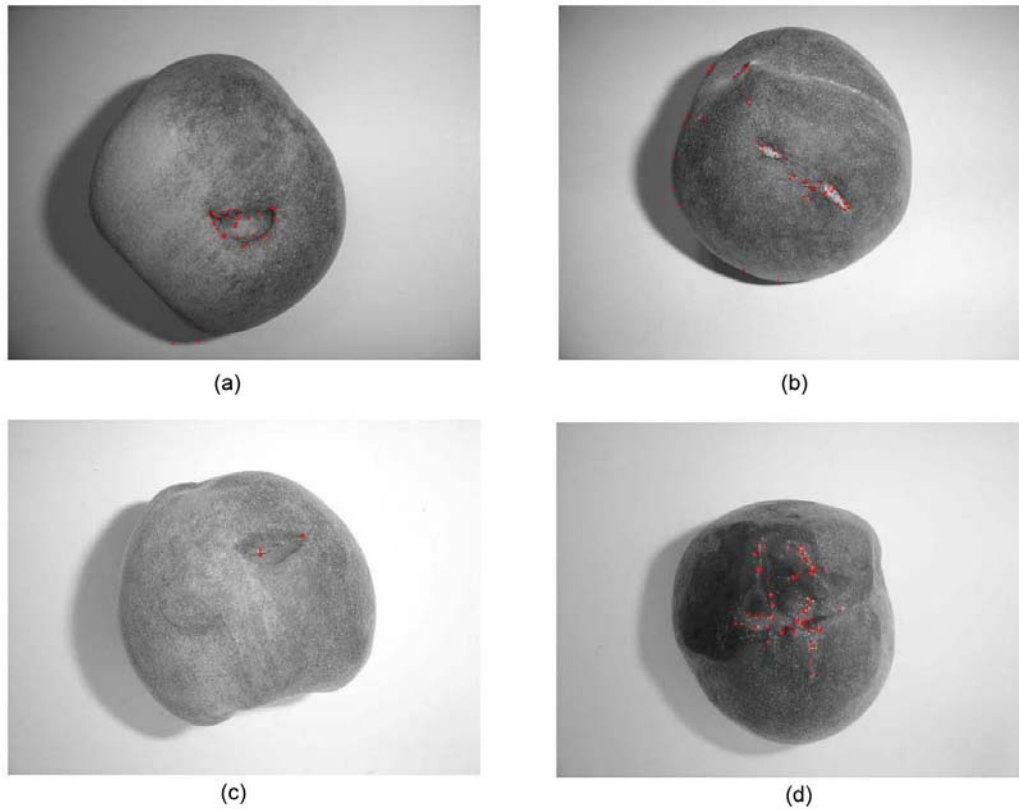


Figura 57 Las imágenes (a), (b), (c) y (d) muestran con cruces rojas las zonas que el selector detecta como defectuosas. Los ejemplos de esta figura son todos de melocotones con pulpa visible.

En este punto del esquema los melocotones ya han sido clasificados según los criterios ya descritos. El algoritmo debe ponerse a prueba con la colección de imágenes tomadas anteriormente y poder así evaluar la capacidad del sistema y su respuesta ante imágenes reales.

5. Resultados y observaciones

En el apartado 3 “Consideraciones previas” se describe la toma de imágenes para formar la base de datos a partir de la cual podremos ejecutar el algoritmo en Matlab® y comprobar los resultados.

Hay dos parámetros que son de interés y devuelve la función “selector”. El primero, devuelve “1” si el melocotón es sano o bien “0” si no lo es. La segunda variable devuelve el tipo de defecto o si es sano. Este segundo parámetro puede valer:

- “picoteado”
- “pulpa visible”
- “golpe”
- “sano”

El resultado se muestra en forma de tabla donde se separa por columnas los distintos valores tenidos en cuenta:

-Iluminación: para cada prueba queda anotada en esta columna el tipo de luz con que se ha tomado la fotografía:

- Tipo1: Poca luz (imagen oscura)
- Tipo2: Muy poca luz
- Tipo3: Imagen tomada con flash

-Id. Pieza (Identificador de pieza): nombre dado a la imagen y usado para ser asignada a una variable en el algoritmo “selector”.

-Variedad: clase del melocotón de la imagen. Puede ser Flavour crest, rojo, May Crest o Candor.

-Resultado: valor de la variable “result”. Es “1” si la fruta es sana y “0” si no lo es.

-Tipo: variable del algoritmo selector que devuelve “pulpa visible”, “golpe”, “sano” o “picoteado en función de lo detectado”.

-Visual: valoración personal de la imagen del melocotón para poder realizar la comparación con el resultado del “selector”.

-Fallos: columna que destaca de forma visual con cruces rojas aquellos resultados que son fallidos.

LUZ	DESCRIPCIÓN
Tipo1	Sombra
Tipo2	Oscura
Tipo3	Flash

LUZ	ID PIEZA	VARIEDAD	RESULTADO	TIPO	VISUAL	FALLOS
Tipo3	1	May Crest	0	"picoteado"	picoteado	
Tipo2	2	May Crest	0	"picoteado"	picoteado	
Tipo3	3	May Crest	0	"pulpa visible"	picoteado	X
Tipo2	4	May Crest	0	"golpe"	golpeado	
Tipo2	5	May Crest	0	"golpe"	golpeado	
Tipo2	6	May Crest	1	"sano"	sano	
Tipo3	7	May Crest	1	"sano"	sano	
Tipo2	9	May Crest	1	"sano"	sano	
Tipo2	10	May Crest	0	"pulpa visible"	pulpaYgolpe	
Tipo2	11	May Crest	1	"sano"	sano	
Tipo2	12	May Crest	1	"sano"	sano	
Tipo3	13	May Crest	0	"picoteado"	sano	X
Tipo2	14	May Crest	1	"sano"	sano	
Tipo3	15	May Crest	1	"sano"	sano	
Tipo3	16	May Crest	0	"pulpa visible"	picotazos	
Tipo2	17	May Crest	1	"sano"	sano	
Tipo3	18	May Crest	1	"sano"	sano	
Tipo3	19	May Crest	0	"pulpa visible"	sano	X
Tipo2	20	May Crest	1	"sano"	sano	
Tipo3	21	May Crest	1	"sano"	sano	
Tipo2	22	May Crest	1	"sano"	sano	
Tipo3	23	May Crest	1	"sano"	sano	
Tipo2	24	May Crest	1	"sano"	picotazos	X
Tipo2	25	Candor	0	"golpe"	golpeado	

Tipo2	26	May Crest	0	"picoteado"	picotazos	
Tipo3	27	May Crest	0	"pulpa visible"	pulpa visible	
Tipo2	28	May Crest	1	"sano"	sano	
Tipo2	29	May Crest	1	"sano"	sano	
Tipo3	30	May Crest	1	"sano"	sano	
Tipo3	31	May Crest	0	"pulpa visible"	pulpa visible	
Tipo2	32	Candor	0	"golpe"	golpeado	
Tipo3	33	May Crest	1	"sano"	sano	
Tipo2	35	May Crest	0	"picoteado"	picotazo	
Tipo3	36	rojo	0	"pulpa visible"	sano	X (*)
Tipo2	37	rojo	0	"golpe"	sano	X (*)
Tipo2	38	rojo	0	"golpe"	sano	X (*)
Tipo2	39	rojo	1	"sano"	sano	
Tipo1	40	rojo	0	"golpe"	golpeado	
Tipo3	41	candor	0	"pulpa visible"	pulpaYgolpe	
Tipo3	42	May Crest	1	"sano"	sano	
Tipo3	43	May Crest	0	"picoteado"	picoteado	
Tipo2	44	May Crest	0	"golpe"	sano	X
Tipo2	45	May Crest	0	"golpe"	sano	X
Tipo3	46	May Crest	0	"picoteado"	picoteado	
Tipo3	47	May Crest	0	"pulpa visible"	picoteado	X
Tipo2	48	May Crest	0	"golpe"	picoteado	X
Tipo2	49	Candor	0	"golpe"	golpeado	
Tipo3	50	Candor	0	"picoteado"	pulpa visible	X
Tipo2	51	May Crest	0	"pulpa visible"	pulpa visible	
Tipo2	52	May Crest	0	"sano"	sano	
Tipo3	53	May Crest	1	"sano"	sano	
Tipo3	54	May Crest	0	"picoteado"	sano	X
Tipo2	55	May Crest	1	"sano"	sano	
Tipo2	56	May Crest	0	"golpe"	golpeado	
Tipo3	57	May Crest	1	"sano"	sano	
Tipo3	58	May Crest	1	"sano"	sano	
Tipo1	59	May Crest	1	"sano"	sano	
Tipo2	60	May Crest	1	"sano"	sano	
Tipo3	61	May Crest	0	"picoteado"	picoteado	
Tipo2	62	May Crest	0	"golpe"	sano	X
Tipo2	63	May Crest	1	"sano"	sano	
Tipo3	64	May Crest	1	"sano"	golpeado	X
Tipo2	65	May Crest	0	"golpe"	golpeado	
Tipo3	66	May Crest	1	"sano"	sano	
Tipo3	67	May Crest	1	"sano"	sano	
Tipo3	68	May Crest	0	"golpe"	golpeado	
Tipo3	69	May Crest	1	"sano"	picoteado	X
Tipo3	70	Candor	0	"golpe"	golpe	
Tipo1	71	May Crest	0	"pulpa visible"	pulpa visible	

Tipo3	72	May Crest	0	"pulpa visible"	picoteado	X
Tipo3	73	May Crest	1	"sano"	sano	
Tipo1	74	May Crest	1	"sano"	sano	
Tipo3	75	May Crest	1	"sano"	sano	
Tipo3	76	May Crest	0	"pulpa visible"	picotazo	
Tipo1	77	May Crest	1	"sano"	sano	
Tipo1	78	May Crest	1	"sano"	sano	
Tipo3	81	Candor	0	"pulpa visible"	sano	X
Tipo3	82	Candor	1	"sano"	sano	
Tipo3	83	Candor	0	"golpe"	golpeado	
Tipo3	85	May Crest	0	"pulpa visible"	pulpa visible	
Tipo3	86	Candor	1	"sano"	sano	
Tipo3	87	Candor	1	"sano"	sano	
Tipo3	88	Candor	1	"sano"	sano	
Tipo1	89	May Crest	0	"golpe"	sano	X
Tipo1	90	May Crest	0	"golpe"	sano	X
Tipo1	91	May Crest	0	"picoteado"	picotazos	
Tipo3	92	May Crest	0	"pulpa visible"	pulpa visible	
Tipo1	93	May Crest	1	"sano"	sano	
Tipo1	94	May Crest	0	"golpe"	golpeado	
Tipo3	95	May Crest	1	"sano"	sano	
Tipo3	96	May Crest	0	"pulpa visible"	pulpa visible	
Tipo1	98	May Crest	1	"sano"	sano	
Tipo1	99	May Crest	0	"pulpa visible"	pulpa visible	
Tipo1	100	May Crest	1	"sano"	sano	
Tipo2	101	May Crest	1	"sano"	sano	
Tipo3	102	May Crest	0	"picoteado"	sano	X
Tipo3	103	May Crest	1	"sano"	sano	
Tipo3	104	May Crest	1	"sano"	sano	
Tipo1	105	May Crest	1	"sano"	picoteado	X
Tipo3	106	May Crest	1	"sano"	sano	
Tipo3	107	May Crest	0	"pulpa visible"	sano	X
Tipo2	108	May Crest	0	"pulpa visible"	sano	X
Tipo3	109	May Crest	1	"sano"	sano	
Tipo3	110	May Crest	1	"sano"	sano	
Tipo3	112	May Crest	1	"sano"	sano	
Tipo3	114	Candor	0	"golpe"	golpeado	
Tipo3	119	May Crest	1	"sano"	sano	
Tipo3	122	Candor	0	"pulpa visible"	golpeado	X
Tipo1	128	May Crest	1	"sano"	sano	
Tipo1	c18	May Crest	1	"sano"	sano	
Tipo1	131	May Crest	1	"sano"	sano	
Tipo1	133	May Crest	1	"sano"	sano	
Tipo2	117	May Crest	0	"picoteado"	picoteado	

Tipo3	121	Candor	1	"sano"	sano	
Tipo3	125	May Crest	1	"sano"	sano	
Tipo3	130	May Crest	0	"picoteado"	picoteado	

LUZ	ID PIEZA	VARIEDAD	RESULTADO	TIPO	VISUAL	FALLOS
Tipo3	picoteado10	rojo	0	"pulpa visible"	picotazos	X
Tipo3	picoteado11	rojo	0	"pulpa visible"	picotazos	X
Tipo3	sano22	rojo	1	"sano"	sano	
Tipo3	uña28	rojo	0	"pulpa visible"	pulpa visible	
Tipo3	uña29	rojo	0	"pulpa visible"	pulpa visible	
Tipo3	uña37	rojo(fondo)	0	"pulpa visible"	pulpa visible	
Tipo3	uña38	rojo	0	"pulpa visible"	pulpa visible	
Tipo3	golpeado10	rojo	1	"sano"	golpeado	X
Tipo2	golpeado2	rojo	0	"golpe"	golpeado	
Tipo2	golpeado20	rojo	1	"sano"	golpeado	X
Tipo3	golpeado4	rojo	1	"sano"	golpeado	X
Tipo3	golpeado5	rojo	1	"sano"	golpeado	X
Tipo2	picoteado1	rojo	0	"picoteado"	picotazo	
Tipo3	picoteado2	rojo	1	"picoteado"	picotazo	
Tipo3	picoteado4	rojo	0	"pulpa visible"	picotazos	X
Tipo2	picoteado5	rojo	0	"picoteado"	picotazo	
Tipo3	picoteado8	rojo	0	"picoteado"	picotazo	
Tipo1	sano13	rojo	1	"sano"	sano	
Tipo3	uña1	rojo	0	"pulpa visible"	pulpa visible	
Tipo3	uña5	rojo	0	"pulpa visible"	pulpa visible	
Tipo2	sano18	rojo	1	"sano"	sano	
Tipo3	sano15	rojo	1	"sano"	sano	
Tipo3	uña2	rojo	0	"pulpa visible"	pulpa visible	
Tipo1	uña6	rojo	0	"pulpa visible"	pulpa visible	

LUZ	ID PIEZA	VARIEDAD	RESULTADO	TIPO	VISUAL	FALLOS
Tipo3	sano_groc	FlavourCrest	1	"sano"	sano	
Tipo3	golpe	FlavourCrest	0	"golpe"	golpeado	
Tipo3	picoteado	FlavourCrest	0	"picoteado"	picotazos	
Tipo3	uña1	FlavourCrest	0	"picoteado"	pulpa visible	X
Tipo3	sano22	FlavourCrest	1	"sano"	sano	
Tipo3	sano23	FlavourCrest	1	"sano"	sano	

() melocotones tipo May Crest con demasiada tonalidad roja más parecidos a la variedad Spring Lady. Podrían excluirse del estudio por no ser un hecho habitual.*

De un total de 146 melocotones, ha habido 33 fallos, un 77,39% de acierto.

ESTUDIO DE LOS RESULTADOS POR GRUPOS

-El primer grupo de frutas está constituido por melocotones de las variedades *Candor* y *May Crest*, ambos de tonalidades rojizas y amarillentas y de piel aterciopelada. Para este grupo:

116 piezas analizadas con el “selector” → 25 resultados erróneos, lo que supone un acierto del 78,45%. Si se excluyen del estudio las tres piezas considerables de otra variedad, el acierto aumenta al 81,03%.

-El segundo grupo está formado por melocotones de la variedad “Rojos”, cuyo tono es más rojizo, casi granate, comparado con la variedad anterior. Para este grupo:

24 piezas analizadas con el “selector” → 7 resultados erróneos, lo que supone un acierto del 70,83%.

-El tercer grupo está formado por melocotones de la variedad *Flavour Crest*. Como la mayoría de las variedades de invierno, la *Flavour Crest* produce melocotones de tamaño inferior y tonalidad parecida a la *May Crest*. Sin embargo esta variedad tiene la piel parecida a las nectarinas, pues prácticamente carece del tacto aterciopelado. Para este grupo:

6 piezas analizadas con el “selector” → 1 resultado erróneo, lo que supone un acierto del 83,33%.

ESTUDIO DE LOS RESULTADOS SEGÚN LA ILUMINACIÓN

Recordando los tres tipos de iluminación con que se han tomado las imágenes, diferenciamos entre ellos para ver cómo afectan al acierto del selector.

-Para el tipo de luz llamada “tipo1”: son imágenes tomadas a la sombra y tienen aspecto apagado. Resultados:

Un total de 19 imágenes han sido tomadas en estas condiciones, resultan erróneas 3. Esto supone un total de acierto del 84,21%.

-Para el tipo de luz llamada “tipo2”: imágenes tomadas sin flash. Más oscuras que el primer tipo.

Se han analizado un total de 42 imágenes con este tipo de luz y ha habido 9 fallos. Se ha acertado un 78,57% de las veces.

-Para el tipo de luz llamada “tipo3”: imágenes tomadas con flash.

De 83 imágenes analizadas, 21 resultan erróneas. Estos datos suponen un acierto del 74,70%.

ESTUDIO DE LOS RESULTADOS SEGÚN CLASIFICACIÓN GENERAL/DEFECTOS

En este apartado se centra el estudio de resultados en el objetivo inicial del proyecto: determinar si el melocotón analizado por el selector es bueno (result=1) o no (result=0), por lo que sólo es necesario observar la columna “resultado” y “fallos”. De un total de 146 imágenes analizadas, han fallado respecto a este criterio 23. Se ha acertado en un 84,25% de las veces.

Los fallos producidos bien han sido por determinar como sana una pieza cuando no lo era como la situación inversa. Los casos más frecuentes han sido

la clasificación de un melocotón como defectuoso “golpeado” en vez de sano o viceversa.

Por otro lado en la clasificación de defectos los errores se producen por la mala diferenciación entre melocotones con desgarros en la piel (defectos de “pulpa visible”) y agujeros de la piel (“picoteado”): de 146 imágenes analizadas, 10 fallos han sido por este tipo de clasificación defectuosa. Se ha acertado un 93,10% de las veces.

Si bien los resultados muestran un porcentaje de acierto favorable, en una aplicación real, dirigida a su comercialización y aplicación industrial el porcentaje general debería situarse en aproximadamente un 80% de acierto en la clasificación general (bueno o malo).

Los resultados más desfavorables han sido obtenidos durante la clasificación según el tipo de luz, de aciertos en un 70% de los casos. El selector ha sido creado con parámetros en la binarización favorables a la luz del tipo3, cosa que hacia previsible errores para cualquier otro tipo de luz. Esta observación carecería de importancia en un sistema de visión en la industria logística de la fruta pues las condiciones de luz en que se toman las imágenes siempre son las mismas.

En las imágenes en niveles de gris con los defectos marcados se pueden apreciar las zonas en que el algoritmo “selector” centra sus cálculos.

RESULTADOS: IMÁGENES REPRESENTATIVAS

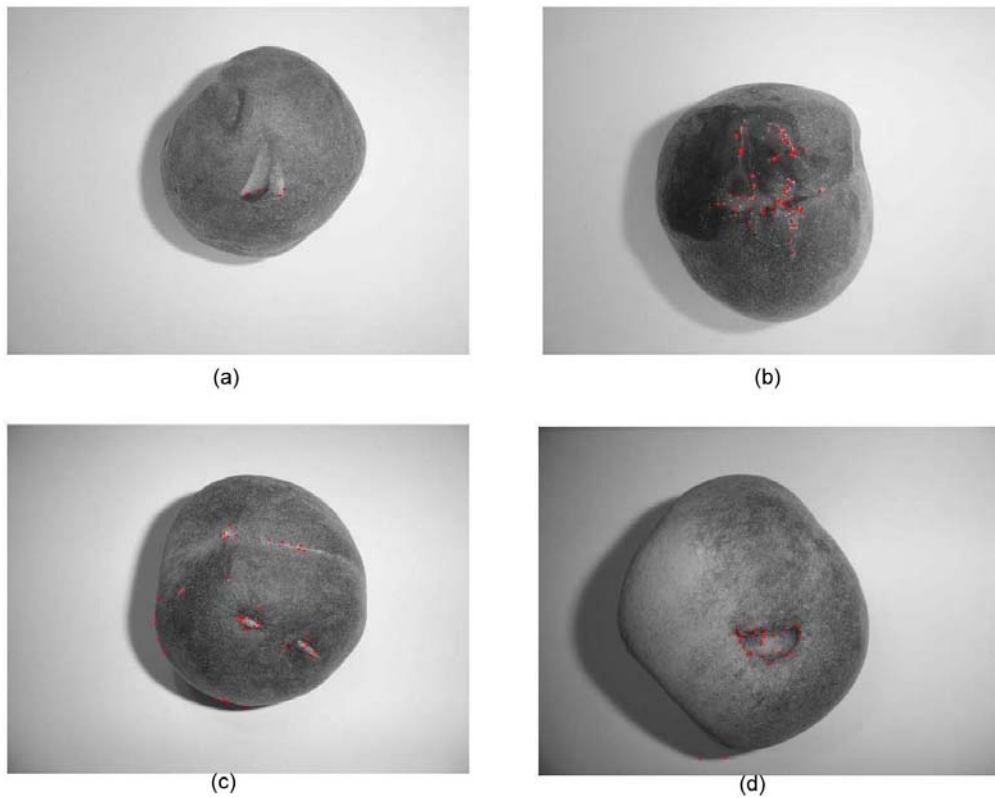


Figura 58 (a) melocotón con desgarro en la piel que deja a la vista la pulpa. (b) melocotón con golpe con pulpa a la vista, (c) Arañazos en la piel y (d) melocotón con pulpa a la vista. Este tipo de defectos aceleran la descomposición de la fruta y interesa su detección. Las pequeñas marcas en rojo marcan las zonas afectadas.

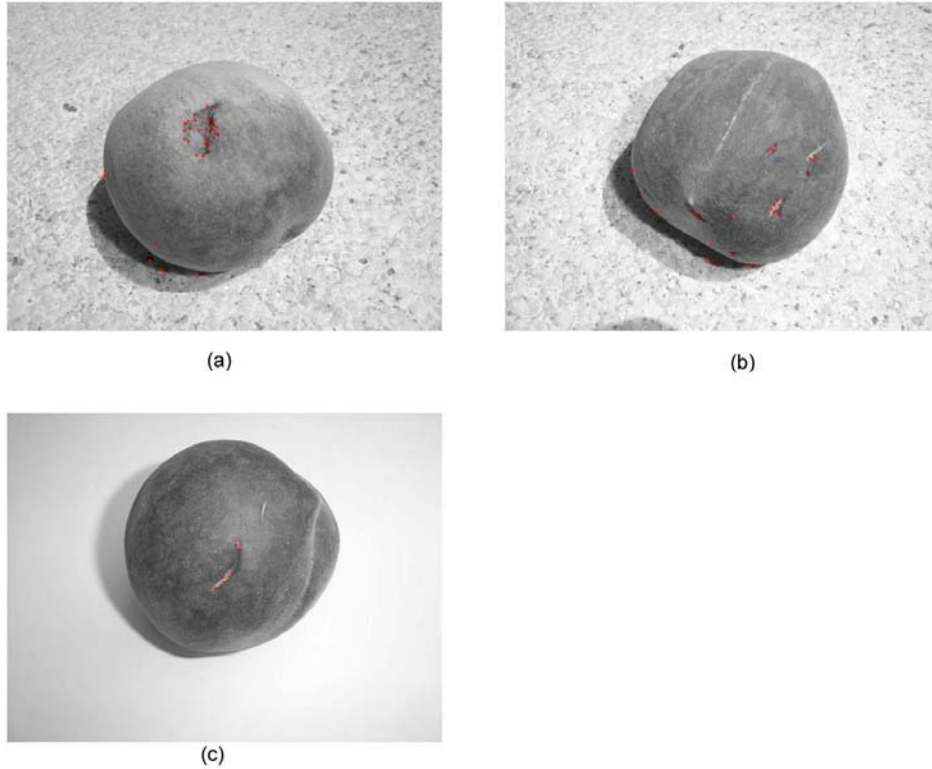
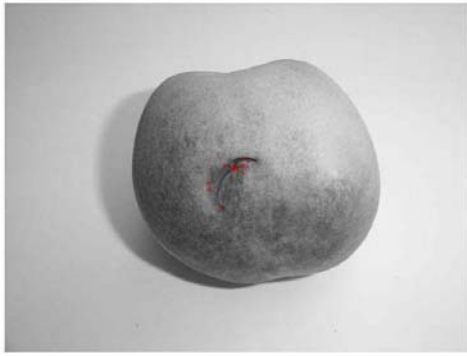


Figura 59 (a) y (b) Muestran dos imágenes de melocotones con fondos con textura y las zonas que el algoritmo destaca como defectuosas marcadas en rojo. Algunas de esas zonas pertenecen a la sombra del melocotón, debido a la textura granulosa del fondo. (c) melocotón de tipo Rojo con los defectos marcados en la zona arañada.



(a)



(b)



(c)

Figura 60 (a) y (b) muestran defectos similares en la piel de la fruta marcados en rojo. (c) Melocotón con un único picotazo.

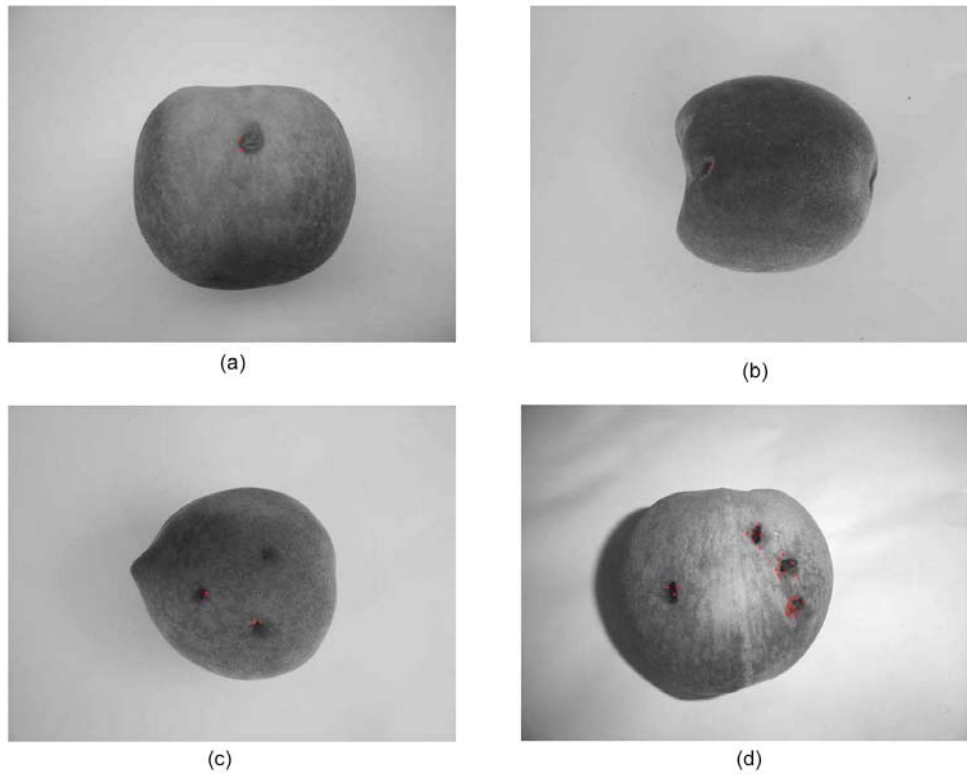


Figura 61 (a) melocotón con un único picotazo. (b) melocotón con un defecto en la parte izquierda. Las frutas (c) y (d) muestran varios agujeros. En la pieza (c) el “selector” sólo detecta dos de los tres defectos.

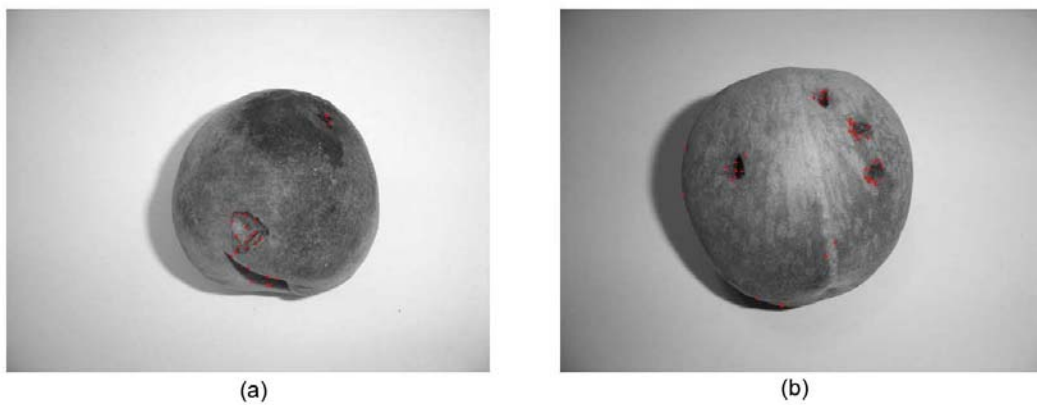


Figura 62 (a) Melocotón con defecto de pulpa visible con las zonas marcadas en rojo. (b) Melocotón picoteado con los defectos marcados en rojo alrededor de los agujeros.

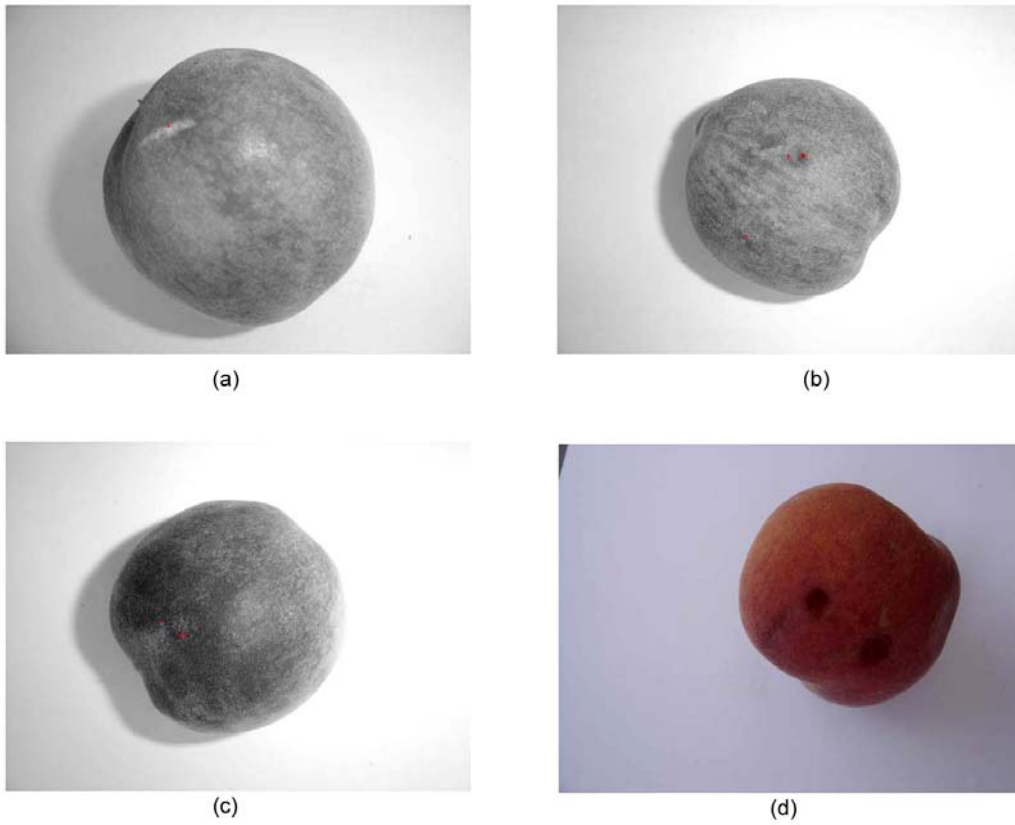
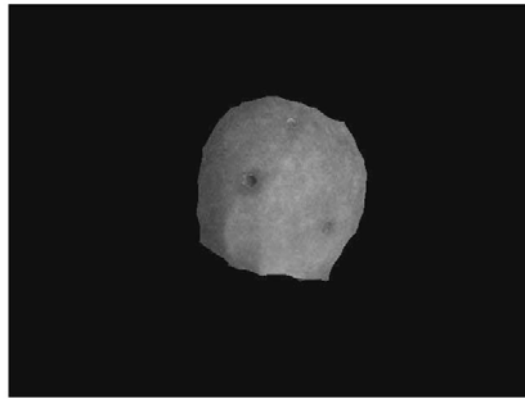
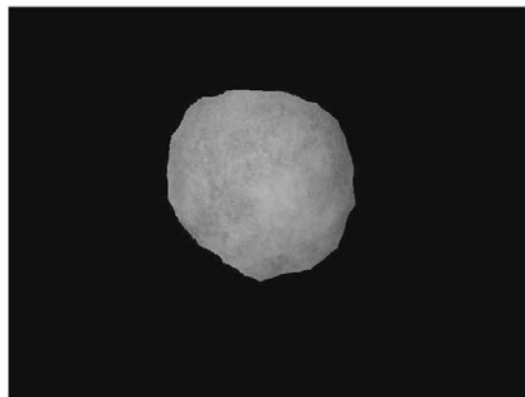


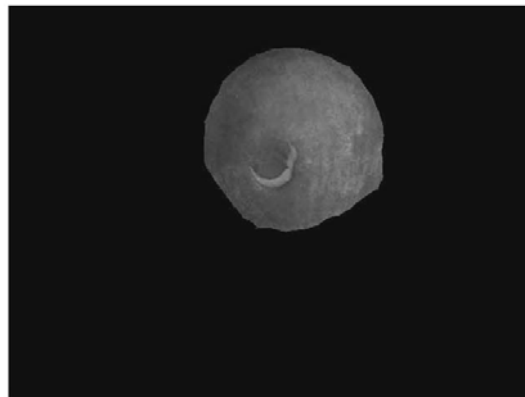
Figura 63 (a) En esta imagen el melocotón sería considerado sano, sin embargo el “selector” detecta un cambio de textura en Sobel en la zona clara y clasifica como malo una fruta que en principio sería sana.



(a)



(b)



(c)

Figura 64 (a) Segmentación de un melocotón picoteado (b) Segmentación de un melocotón sano (c) Segmentación de un melocotón con defecto tipo “pulpa visible”.

6. Conclusiones

Las dificultades de este proyecto han sido muchas, como muchas han sido también las decisiones que se han ido tomando. Ya desde el inicio está claro que es un estudio en que deben realizarse muchas pruebas y se debe fijar un objetivo.

El objetivo del estudio es crear un programa con Matlab® que clasificara las imágenes de fruta de entrada (melocotones) en buenas o malas. En este punto las primeras dificultades se hacen visibles: qué fruta escoger y cómo determinar si es o no apta para el consumo directo. El primer punto se resolvió decidiéndose por melocotones ya que estos, por su textura y coloración, resultaban más complicados e interesantes que no las manzanas o naranjas, de coloración más definida. El segundo punto es absolutamente visual: una persona es capaz de decir por el aspecto si la fruta esta o no en buen estado.

Los resultados del algoritmo selector están determinados en gran medida por dos aspectos: variedad de la fruta e iluminación. La variedad de la fruta resulta un problema a tener en cuenta a la hora de determinar los umbrales de binarización. No es lo mismo una fruta de tonos amarillentos que una de tonos rojizos.

Inicialmente la fruta con que se trabajó fue Flavour Crest, con aspecto de nectarina y los umbrales de binarización y el procedimiento en sí estuvo basado desde el principio en esa variedad. El problema se encuentra en que las variedades cambian en los centros de compra, por lo que un tipo de melocotón que se encuentra en febrero en el supermercado ya no se encuentra en mayo, puesto que es una fruta de primavera-verano y en cualquier otro mes es importada de Estados Unidos. Por ese motivo, cuando el “selector” estuvo finalizado, ya no había Flavour Crest en ningún lado para hacer más pruebas. Hizo falta entonces recurrir a otras variedades similares: Spring Lady, Candor y May Crest. Además se hicieron pruebas con melocotón de la variedad “Rojo”, cuyos resultados se encuentran en el apartado anterior.

Cabe destacar que todas las cifras y los estudios se basan en variedades concretas de melocotón, por lo que los histogramas se distribuirán de forma diferente y darán otro tipo de información si la variedad de fruta escogida fuera otra.

La iluminación juega un papel muy importante en todos los sistemas de reconocimiento de imágenes. En el caso del selector se comprueba que su grado de acierto se ve claramente afectado por ese parámetro. Asimismo, como ya se ha comentado, el aspecto de la iluminación no debería ser preocupante pues para la función del sistema en una cadena de una empresa las condiciones de luz serían invariantes y los parámetros en los que influye (tales como los umbrales de la binarización) podrían ser fijados más cómodamente.

El fondo de las imágenes ha sido un fondo simple excepto algunos ejemplos en que era una textura clara y sencilla. El motivo de escoger un fondo no complicado es debido a que en una aplicación real la cabina donde se captan las imágenes de los melocotones es siempre la misma y el fondo es una superficie lisa y simple.

Un aspecto que se ha descartado para este estudio es la tridimensionalidad de la fruta, que se solventa aceptando el hecho de que las imágenes de un melocotón son captadas desde diferentes ángulos y procesadas por separado para dar una clasificación final. Del mismo modo se supone que el melocotón circula a través de la cabina a velocidad constante y que los rodillos de transporte permiten que el melocotón esté situado siempre en la misma posición, tal como se especifica para algunas de estas máquinas. También es necesario especificar que las imágenes no necesitan de ningún tratamiento previo aparte del escalado puesto que en un selector real no se aplicaría ningún tipo de corrección artística que ralentice la cadena.

A pesar de que el número de aciertos por el sistema “selector” es bastante satisfactorio, el tiempo de ejecución o respuesta del algoritmo está en una

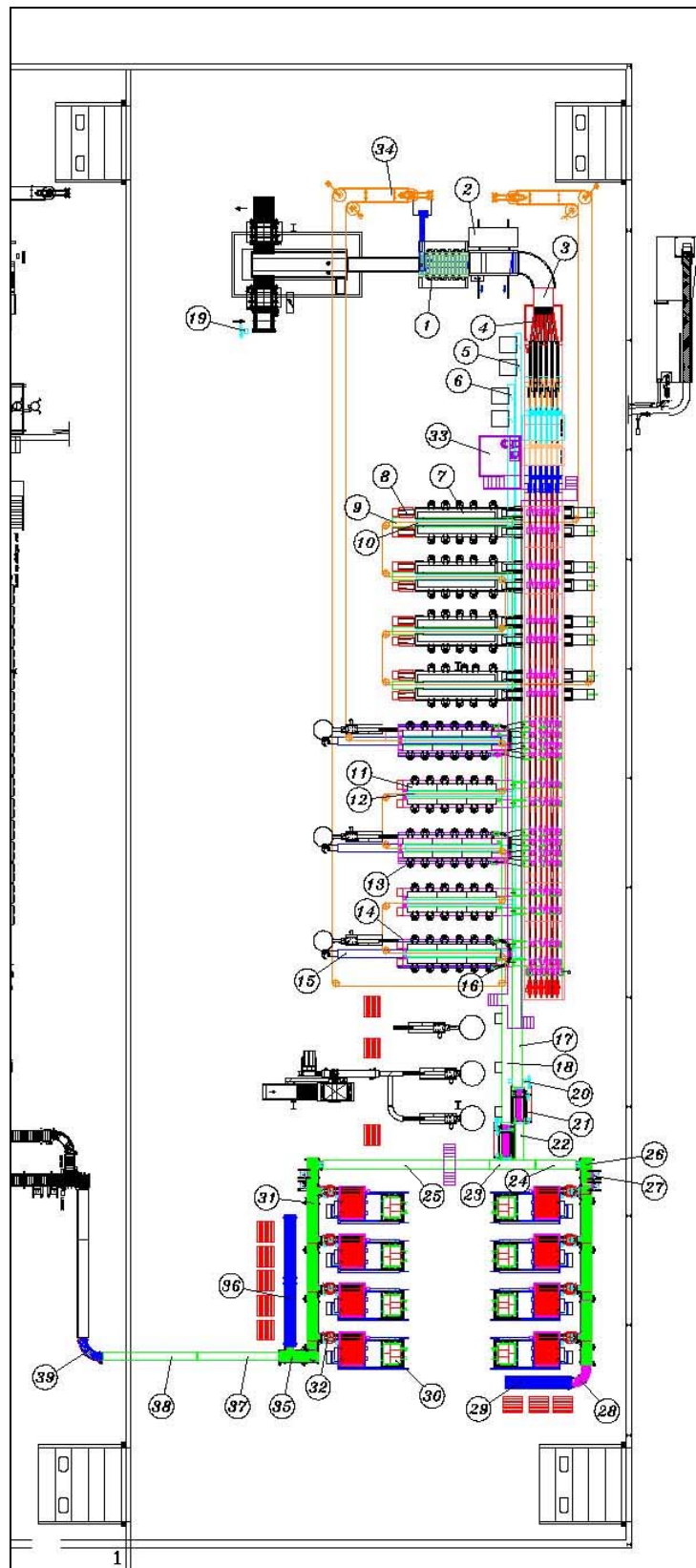
media de 69,5 segundos por imagen. Este resulta un tiempo excesivo si se tiene en cuenta que una aplicación así está pensada para trabajar a tiempo real y a gran velocidad para poder analizar el mayor número de piezas posible en el menor tiempo posible.


Son diversas las mejoras a aplicar al “selector”, que requieren tiempo y dedicación. La disminución del tiempo de ejecución del algoritmo sería una de ellas como también lo sería la reducción del número de fallos en la clasificación general. Asimismo deberían estudiarse otros espacios de color y otras técnicas de procesamiento digital de imágenes que dieran respuesta más eficaz a la detección y clasificación de fallos.

Finalmente queda decir que aunque Matlab® resulte una herramienta especialmente útil y muy visual para desarrollar estudios de este tipo no sería un programa adecuado para la ejecución de un programa de este tipo en un caso real.

7. Anexos

7.1 Anexo I : Situación del algoritmo “selector” en la cadena logística.



MAQUINARIA QUE INTEGRA EL PRESUPUESTO			
—	VOLCADOR DE PALOTS NORIA EN AGUA (RUBIES)		1
—	SOFTWARE DE CONTROL DE PRODUCCION PARA INSTALACION EXISTENTE		1
—	SOFTWARE DE CONTROL DE PRODUCCION PARA INSTALACION NUEVA		1
—	RED DE DATOS MOD. 65RD 9 NODOS L=188M		1
—	SOFTWARE DE TRAZABILIDAD EN PALOT MOD. 65ST		1
—	SOFTWARE DE TRAZABILIDAD EN PALET MOD. 65ST		1
—	SOFTWARE DE TRAZABILIDAD EN CAJA MOD. 65ST		1
—	SOFTWARE DE DISEÑO DE ETIQUETAS EN PC MOD. 02DE (PC INCLUIDO)		1
39	CURVA 90' MOD. S93		1
38	TRANSPORTADOR MOD. 21L5 L=6M (1T)		1
37	TRANSPORTADOR ELEVADOR MOD. 21L5 L=5,5M (1T)		1
36	TRANSPORTADOR DE GRAVEDAD MOD. 20GR L=8M		1
35	TRANSPORTADOR DISTRIBUIDOR MOD. 20RC L=2,5M (2T) (2CS)		1
34	AERED DE CUERDA PARA ALIMENTACION DE CAJAS VACIAS		1
33	PASARELAS DE ACCESO Y CABINA		1
32	GIRADOR DE CAJAS EN PALETIZADOR MOD. 10GP		30
31	TRANSPORTADOR DISTRIBUIDOR MOD. 20RC L=12M (5T) (5CS)		1
30	EUROPALETIZADOR UNIVERSAL MOD. 10ML		8
29	TRANSPORTADOR DE GRAVEDAD MOD. 20GR L=4M		1
28	CURVA 90' GRAVEDAD MOD. 20CG		1
27	ETIQUETADORA TIEMPO REAL MOD. 02BT		4
26	TRANSPORTADOR DISTRIBUIDOR MOD. 20RC L=13M (6T) (5CS)		1
25	TRANSPORTADOR MOD. 21L6 L=10M (1T)		1
24	TRANSPORTADOR MOD. 21L5 L=2,5M (1T)		1
23	TRANSPORTADOR MOD. 21L6 L=3M (1T) (1 CRUCE)		2
22	TRANSPORTADOR MOD. 21L6 L=2,5M (1T)		2
21	TRANSPORTADOR ABATIBLE MOD. 21TA L=2,5M		2
20	SISTEMA DE IDENTIFICACION DE CAJAS MOD. 03LC		8
19	SISTEMA DE IDENTIFICACION DE PALOTS MOD. 03LP		1
18	TRANSPORTADOR MOD. 21L6 L=23,5M (3T) (4 CRUCES) (3 PUESTOS)		1
17	TRANSPORTADOR MOD. 21L6 L=34,5M (4T) (3 CRUCES)		1
16	DOBLE CURVA DE CHARNELA		3
15	TRANSPORTADOR MOD. 21L4 L=10M (1T)		3
14	TRANSPORTADOR MOD. 21L2 L=6M (1T)		6
13	MESA DE CONFECCION DE 4 CAUBRES MOD. 21M5 L=6M		2
12	TRANSPORTADOR MOD. 21L3 L=6,5M (1T)		5
11	MESA DE CONFECCION MOD. 21M5 L=6M		3
10	TRANSPORTADOR MOD. 21L3 L=5,5M (1T)		4
9	TRANSPORTADOR MOD. 21L5 L=7,5M (1T)		4
8	LLENADORA DE CAJAS MANUAL A GRANEL		8
7	TRAY PACK Y MESA DE CONFECCION L=6M		4
6	TRANSPORTADOR MOD. 21L4 L=18M (1T)		1
5	TRANSPORTADOR MOD. 21L4 L=38M (3T)		1
4	CAUBR. MOD. ROLLERSTAR 6/22+1 MARCA AW. (PESO+COLOR+DIAMETRO)		1
3	TRANSPORTADOR MOD. 21L12 L=1,5M (1T)		1
2	CEPILLADORA CON BY-PASS Y CURVA PARA PRODUCTO		1
1	MESA DE TRIA L=3.000mm A=1.200mm		1
Marca Denominación y observaciones			Nº de Maquinas
Proyect	FECHA	NOMBRE	Nº PLANO
Comprab	14/03/08	Carmen	
"ACTEL"			3.24
 SORFRUIT Pol. d'Obradors 6/Llanterna, 9 Tel (98) 3901314 Fax (98) 3901851 46110 CODBELLA (Valencia)			ESCALA:
TECNOLOGIA PARA LOS FRUTOS			1/200

ESTE PLANO ES DE NUESTRA PROPIEDAD, NO PUDIENDO SER COPIADO NI REPRODUCIDO, SIN NUESTRO CONSENTIMIENTO

Los dos planos anteriores han sido cedidos por cortesía de la empresa *Grup ACTEL*, cooperativa que cubre varios servicios en el sector de la fruta y tiene sede en Lleida.

El primer plano corresponde a la cadena logística. Por el número diecinueve marcado en el esquema entran las cajas grandes o contenedores de fruta para luego pasar por el número cuatro que es el aparato reconocedor de color, tamaño y peso. Para este ejemplo el número cuatro correspondería al “selector”, pero con más funciones añadidas aparte de la detección de defectos (en el apartado color). Es interesante remarcar que los colores azul turquesa y marrón claro de la salida del detector (numero 4 en rojo) se corresponden a la selección de calidad y cálculo de los niveles de azúcar respectivamente.

Es destacable la cantidad de transportadores necesarios y la función del elemento marcado con el número 7. Éste se corresponde a la mesa de llenado manual de cajas a granel, donde las trabajadoras recogen una a una las piezas de fruta y las ponen en cajas que luego serán transportadas en camiones hacia los centros de consumo o bien hacia cámaras frigoríficas de la propia empresa.

En la leyenda adjuntada se desglosa la información sobre la cadena y se nombra las marcas de los fabricantes de los aparatos. Obsérvese que el sistema incorpora dos reconocedores de imágenes: uno para la fruta (número 4) y otro para las cajas de entrada (número 19).

7.2 Anexo II : Sistemas de adquisición en la cadena logística.

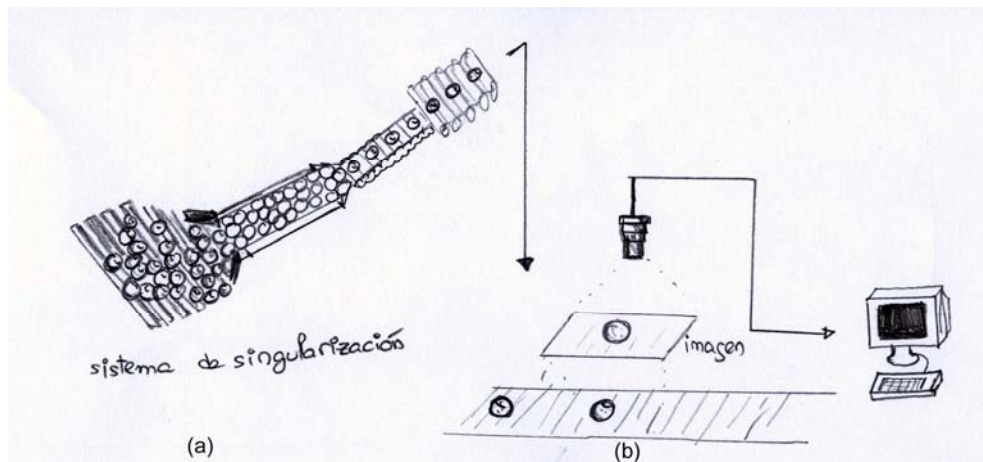


Figura 65 (a) Sistema de singularización mediante el cual los melocotones circulan de uno en uno por la cadena transportadora hasta la zona de análisis. (b) Sistema de captación de una única cámara conectada a un PC.

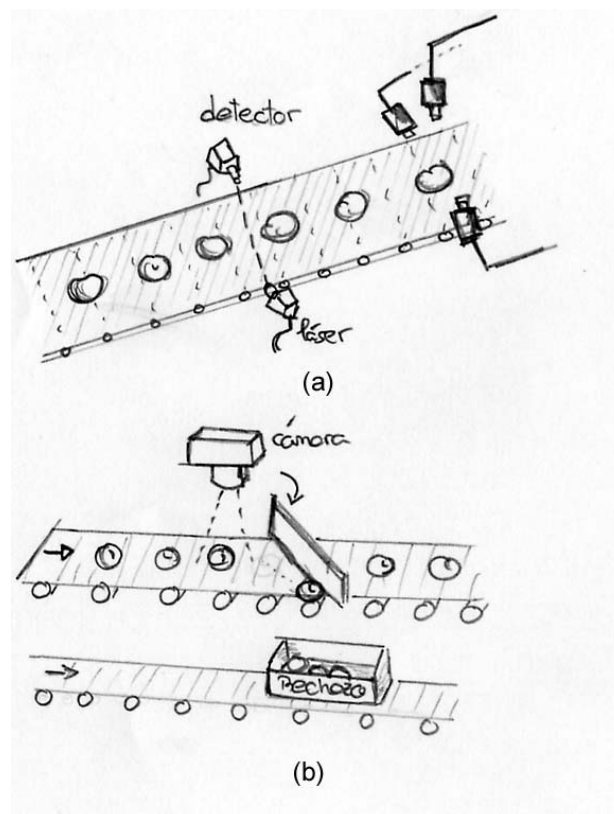


Figura 66 (a) Sistema de adquisición multicámara. Este transportador incorpora un sistema contador de piezas de fruta. (b) Sistema con una única cámara.

Una palanca se activa para desviar las piezas rechazadas y enviarlas a otro punto de la cadena que seguirá un recorrido distinto.

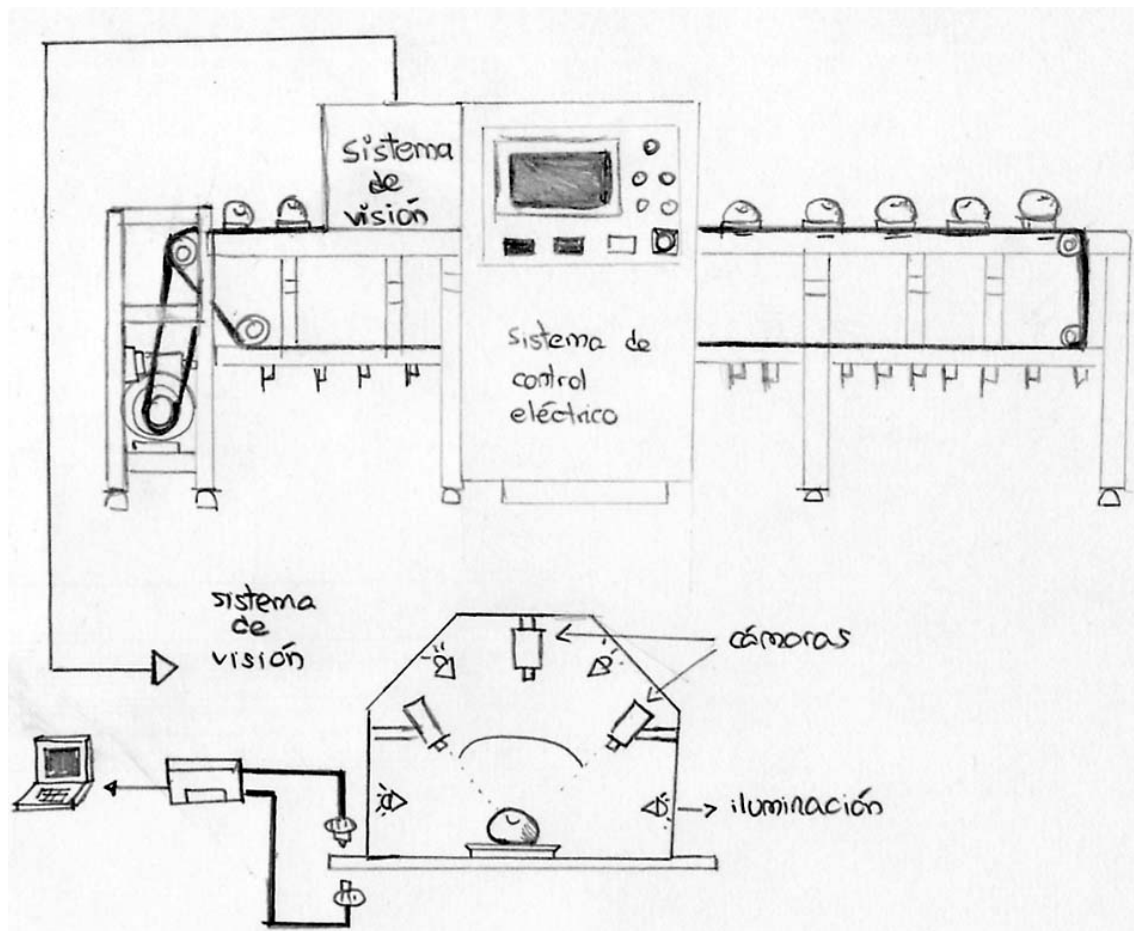


Figura 67 Sistema simple formado por la cadena transportadora, una cabina de control y acceso mecánico-eléctrico. Incorpora un sistema de visión que incluye una cabina con tres cámaras, puntos de luz y un sistema sensor de fibra óptica.

3. Referencias bibliográficas

[1] GONZALEZ, R.C; WOODS, R.E , “Digital Image Processing”, Addison-Wesley Publishing Company, pp 413-415, 1993

[2] GONZALEZ, R.C; WOODS, R.E , “Digital Image Processing” , Addison-Wesley Publishing Company, pp 443-444, 1993

[3] www.vilaweb.cat , sección “noticias”, jueves 09/03/06 6:00 h

Bibliografía consultada

GONZALEZ, R.C; WOODS, R.E , “Digital Image Processing” , Addison-Wesley Publishing Company, pp 199,419,479,413-415,443-444, 1993

NJOROGE, J.B., NINOMIYA, K., KONDO, N., TOITA, H., “Automated fruit grading system using image processing”, *SICE 2002. Proceedings of the 41st SICE Annual Conference*, volum 2, pp 1346–1351, 5-7 Agost 2002

GUO F., CAO Q. “Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on Control and Automation”, volum 6, p 4802 – 4805, 15-19 Juny 2004.

RECCE M., TAYLOR J.” High speed vision-based quality grading of oranges”. Neural Networks for Identification, Control, Robotics, and Signal/Image Processing. Proceedings., International Workshop on, 21-23 , pp 136-144, 1996.

BULANON, D.M. KATAOKA, T. OKAMOTO, H. HATA, S. “Development of a real-time machine vision system for the apple harvesting robot”, *SICE 2004 Annual Conference*, volum 1, pp 595 – 598, 2004

CARBO PERICAY J., IGLESIAS CASTELLARNAU I, “Presseguer. Les varietats de més interés”, Institut de Recerca i Tecnologia Agroalimentàries, Generalitat de Catalunya. 2005

CIELO, P “Optical Techniques for Industrial Inspection”, Academic Press, Inc, pp 257-307, 1998.

Documentación consultada en Internet

www.vilaweb.cat

ieeexplore.ieee.org

biblioteca.upc.es/bib170/

en.wikipedia.org/wiki/HSV_color_space

[en.wikipedia.org/wiki/thresholding_\(image_processing\)](http://en.wikipedia.org/wiki/thresholding_(image_processing))

http://www.hiru.com/es/matematika/matematika_05900.html

<http://www.aulafacil.com/CursoEstadistica/Lecc-4-est.htm>

<http://aldocgh.tripod.com/>

<http://ceres.ugr.es/~amp/docencia/tepds/pract5.pdf>

<http://www.bioestadistica.uma.es/libro/node16.htm>

<http://www.quesabesde.com/camdig/articulos.asp?articulo=117>

<http://www.caborian.com/content/view/380/152/>

<http://informatica.uv.es/iiguia/2000/VC/practica5-04-05.pdf>

http://www.ruf.rice.edu/~lane/stat_sim/descriptive/index.html

<http://scien.stanford.edu/class/psych221/projects/02/sojeong/#basic>

<http://www.viarural.com.ar/viarural.com.ar/agroindustria/frutas-empaque/unitec/default.html>

http://www.dfmf.uned.es/actividades/no_reglada/laboratorio/imagen_binaria.pdf.